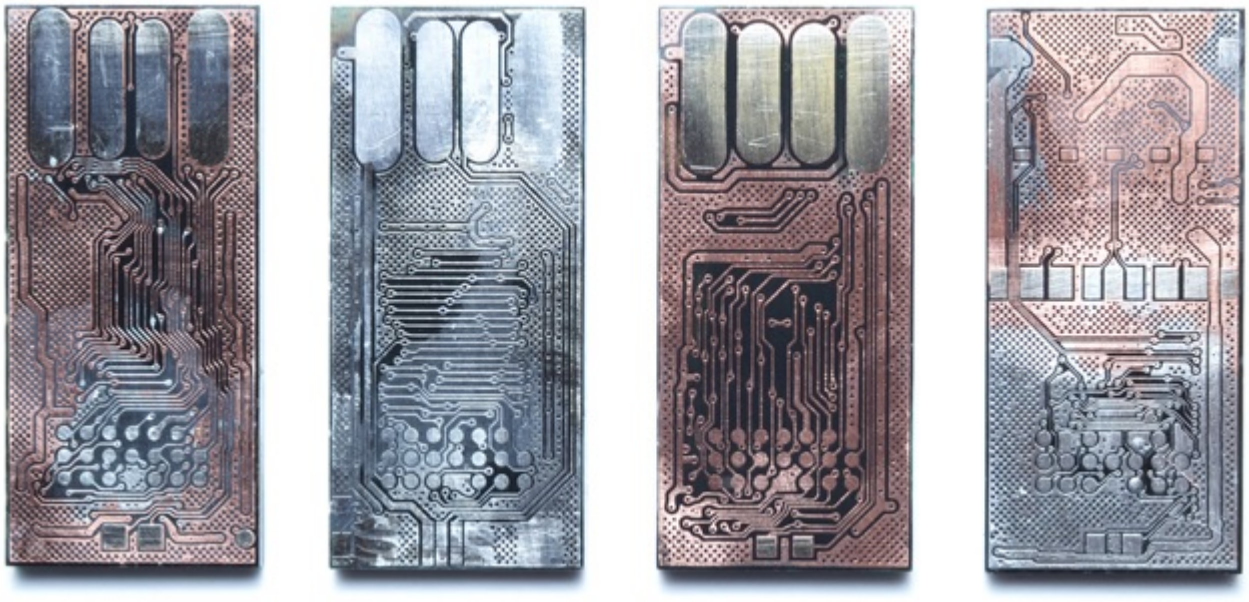
Help Center

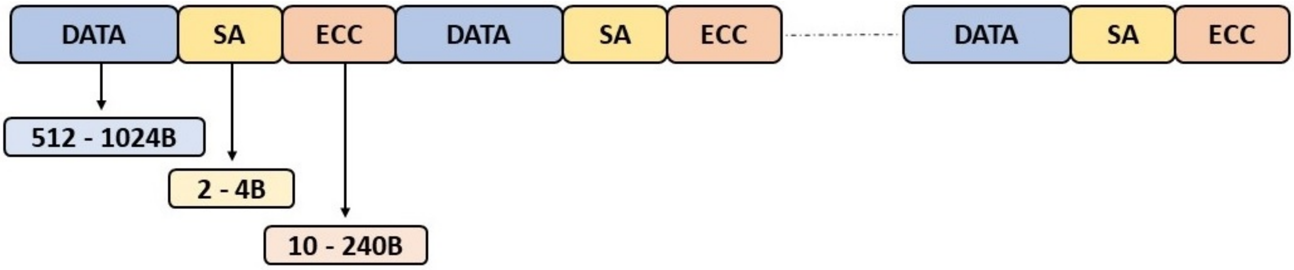
Alcor Micro(AU) controllers - Peculiarities of data recovery

Alcor Micro (AU) controllers are found in all sorts of devices such as microSD cards, USB flash drives, SD cards of any package, and especially monolithic devices. They are very popular in cheap, refurbished, and fake devices as well. Recent models do not vary a lot compared to other controller brands. This article is an attempt to summarize and generalize the chip-off data recovery process from devices with such controllers, especially AU6989 and AU6998 family.



Page structures

In the first step of each data recovery procedure is necessary to determine page layout. Every controller has its unique page structure, in the case of Alcor Micro, it presents as follows.

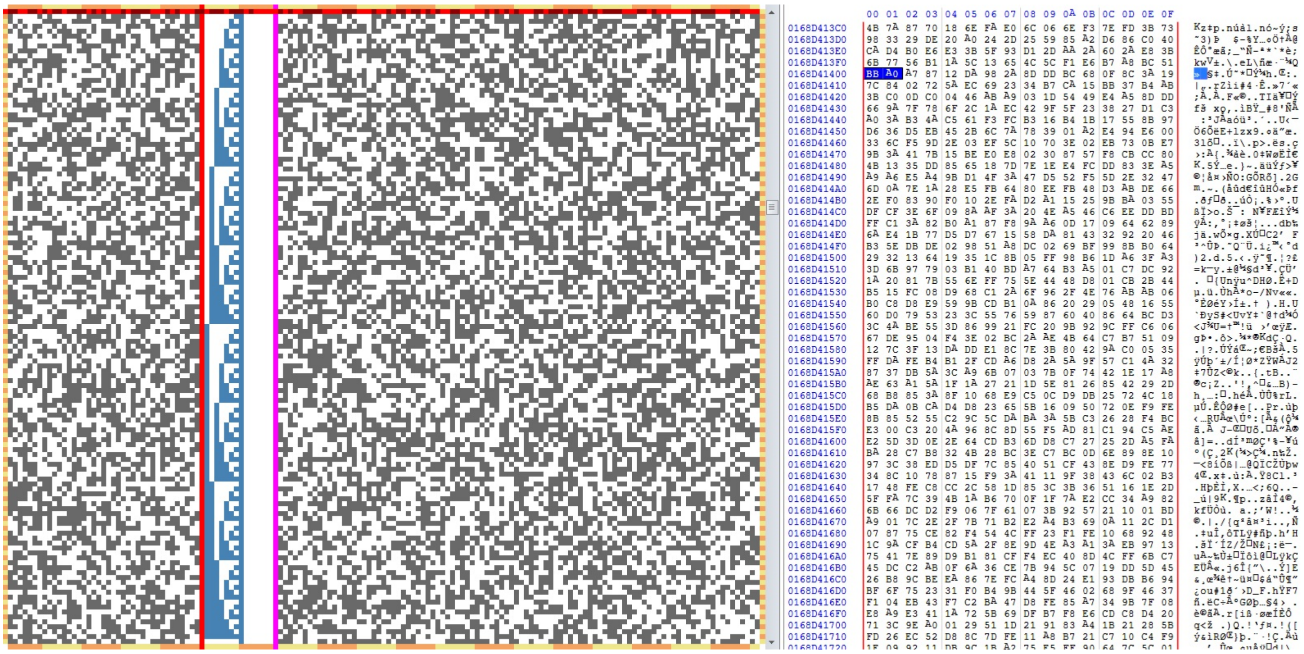


Data area in the majority cases with AU controllers is 1024 Bytes, we can spot as well 512 Byte data area which commonly occurs in low capacity devices.

ECC area, like in every case, can vary from 10 to 240 Bytes and its highly dependable from controller model and manufacturer settings. Usually, AUs are reserving 70, 77, 120, or 126 Bytes for ECC codes.

Service area may have 2,3 or up to 4 Bytes. The most popular, especially for AU6989 controllers, is 2 Bytes and in the case of 2 Byte Service area, it also may be XORed.

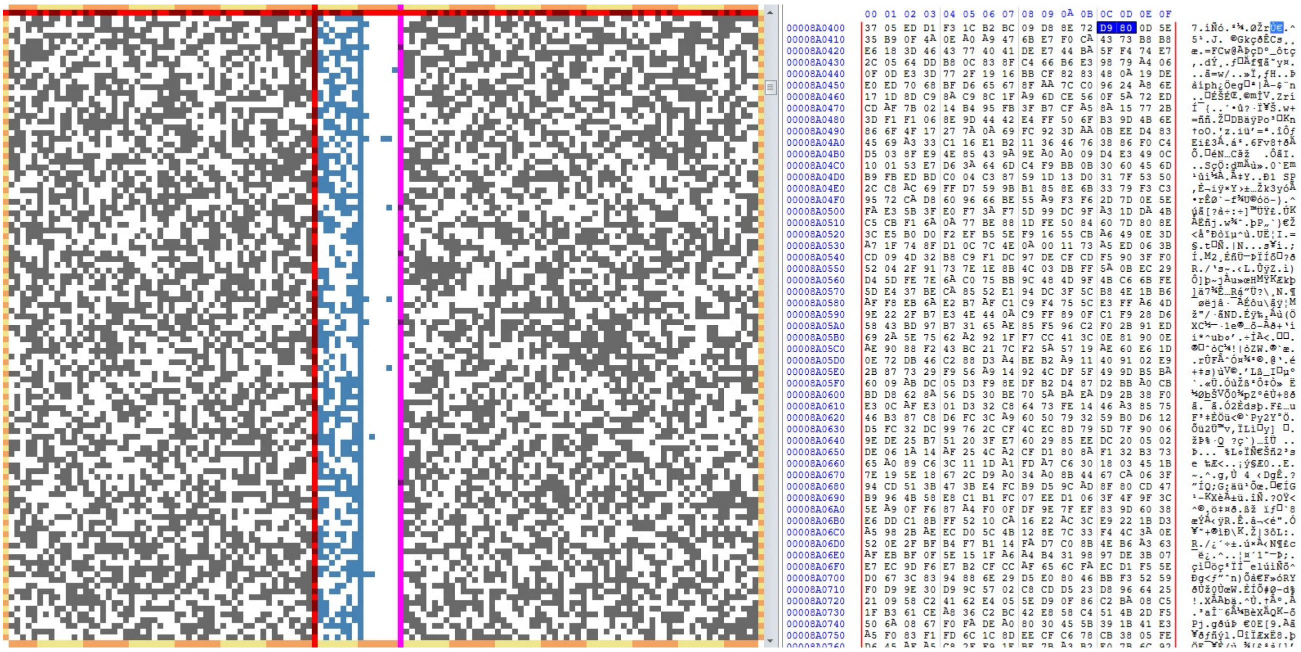
- **Non - XORed Service area**



```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0168D413C0  4B 7A 87 70 18 6E FA E0 6C 06 6E F3 7E FD 3B 73
0168D413D0  98 33 29 DE 20 A0 24 2D 25 59 85 A2 D6 86 C0 40
0168D413E0  CA D4 B0 E6 E3 3B 5F 93 D1 2D AA 2A 60 2A E8 3B
0168D413F0  6B 77 56 B1 1A 5C 13 65 4C 5C F1 E6 B7 A8 BC 51
0168D41400  BB A0 A7 87 12 DA 98 2A 8D DD BC 68 0F 8C 3A 19
0168D41410  7C 84 02 72 5A EC 69 23 34 B7 CA 15 BB 37 B4 AB
0168D41420  3B C0 0D C0 04 46 AB A9 03 1D 54 49 E4 A5 8D DD
0168D41430  66 9A 7F 78 6F 2C 1A EC 42 9F 5F 23 38 27 D1 C3
0168D41440  A0 3A B3 4A C5 61 F3 FC B3 16 B4 1B 17 55 8B 97
0168D41450  D6 36 D5 EB 45 2B 6C 7A 78 39 01 A2 E4 94 E6 00
0168D41460  33 6C F5 9D 2E 03 EF 5C 10 70 3E 02 EB 73 0B E7
0168D41470  9B 3A 41 7B 15 BE E0 E8 02 30 87 57 F8 CB CC 80
0168D41480  4B 13 35 DD 85 65 18 7D 7E 1E E4 FC DD 83 3E A5
0168D41490  A9 A6 E5 A4 9B D1 4F 3A 47 D5 52 F5 5D 2E 32 47
0168D414A0  6D 0A 7E 1A 28 E5 FB 64 80 EE FB 48 D3 AB DE 66
0168D414B0  2E F0 83 90 F0 10 2E FA D2 A1 15 25 9B BA 03 55
0168D414C0  DF CF 3E 6F 09 8A AF 3A 20 4E A5 46 C6 EE DD BD
0168D414D0  FF C1 3A 82 B0 A1 87 F8 9A A6 0D 17 09 64 62 89
0168D414E0  6A E4 1B 77 D5 D7 67 15 58 DA 81 43 32 92 20 46
0168D414F0  B3 5E DB DE 02 98 51 A8 DC 02 69 BF 99 8B B0 64
0168D41500  29 32 13 64 19 35 1C 8B 05 FF 98 B6 1D A6 3F A3
0168D41510  3D 6B 97 79 03 B1 40 BD A7 64 B3 A5 01 C7 DC 92
0168D41520  1A 20 81 7B 55 6E FF 75 5E 44 48 D8 01 CB 2B 44
0168D41530  B5 15 FC 08 D9 68 C1 2A 6F 96 2F 4E 76 AB AB 06
0168D41540  B0 C8 D8 E9 59 9B CD B1 0A 86 20 29 05 48 16 55
0168D41550  60 D0 79 53 23 3C 55 76 59 87 60 40 86 64 BC D3
0168D41560  3C 4A BE 55 3D 86 99 21 FC 20 9B 92 9C FF C6 06
0168D41570  67 DE 95 04 F4 3E 02 BC 2A AE 4B 64 C7 B7 51 09
0168D41580  12 7C 3F 13 DA DD E1 8C 7E 3B 80 42 9A C0 05 35
0168D41590  FF DA FE B4 B1 2F CD A6 D8 2A 5A 9F 57 C1 4A 32
0168D415A0  87 37 DB 5A 3C A9 6B 07 03 7B 0F 74 42 1E 17 A8
0168D415B0  AE 63 A1 5A 1F 1A 27 21 1D 5E 81 26 85 42 29 2D
0168D415C0  68 B8 85 3A 8F 10 68 E9 C5 0C D9 DB 25 72 4C 18
0168D415D0  B5 DA 0B CA D4 D8 23 65 5B 16 09 50 72 0E F9 FE
0168D415E0  8B 85 52 55 C2 9C 5C DA BA 3A 5B C3 26 28 F4 BC
0168D415F0  E3 00 C3 20 4A 96 8C 8D 55 F5 AD 81 C1 94 C5 AE
0168D41600  E2 5D 3D 0E 2E 64 CD B3 6D D8 C7 27 25 2D A5 FA
0168D41610  BA 28 C7 B8 32 4B 28 BC 3E C7 BC 0D 6E 89 8E 10
0168D41620  97 3C 38 ED D5 DF 7C 85 40 51 CF 43 8E D9 FE 77
0168D41630  34 8C 10 78 87 15 F9 3A 41 11 9F 38 43 6C 02 B3
0168D41640  17 48 FE C8 CC 2C 58 1D 85 3C 3B 36 51 16 1E 2D
0168D41650  5F FA 7C 39 4B 1A B6 70 0F 1F 7A E2 CC 34 A9 82
0168D41660  6B 66 DC D2 F9 06 7F 61 07 3B 92 57 21 10 01 BD
0168D41670  A9 01 7C 2E 2F 7B 71 B2 E2 A4 B3 69 0A 11 2C D1
0168D41680  07 87 75 CE 82 F4 54 4C FF 23 F1 FE 10 68 92 48
0168D41690  1C 9A CF B4 CD 5A 2F 8E 9D 4E A3 A1 3A EB 97 13
0168D416A0  75 41 7E 89 D9 B1 81 CF F4 EC 40 8D 4C FF 6B C7
0168D416B0  45 DC C2 AB 0F 6A 36 CE 7B 94 5C 07 19 DD 5D 45
0168D416C0  26 B8 9C BE EA 86 7E FC A4 8D 24 E1 93 DB B6 94
0168D416D0  BF 6F 75 23 31 F0 B4 9B 44 5F 46 02 68 9F 46 37
0168D416E0  F1 04 EB 43 F7 C2 BA 47 D8 FE 85 A7 34 9B 7F 08
0168D416F0  E8 A9 E3 41 1A 72 5B 69 DF B7 F8 E6 CD C8 D4 20
0168D41700  71 3C 9E A0 01 29 51 1D 21 91 83 A4 1B 21 28 5B
0168D41710  FD 26 EC 52 D8 8C 7D FE 11 A8 B7 21 C7 10 C4 F9
```

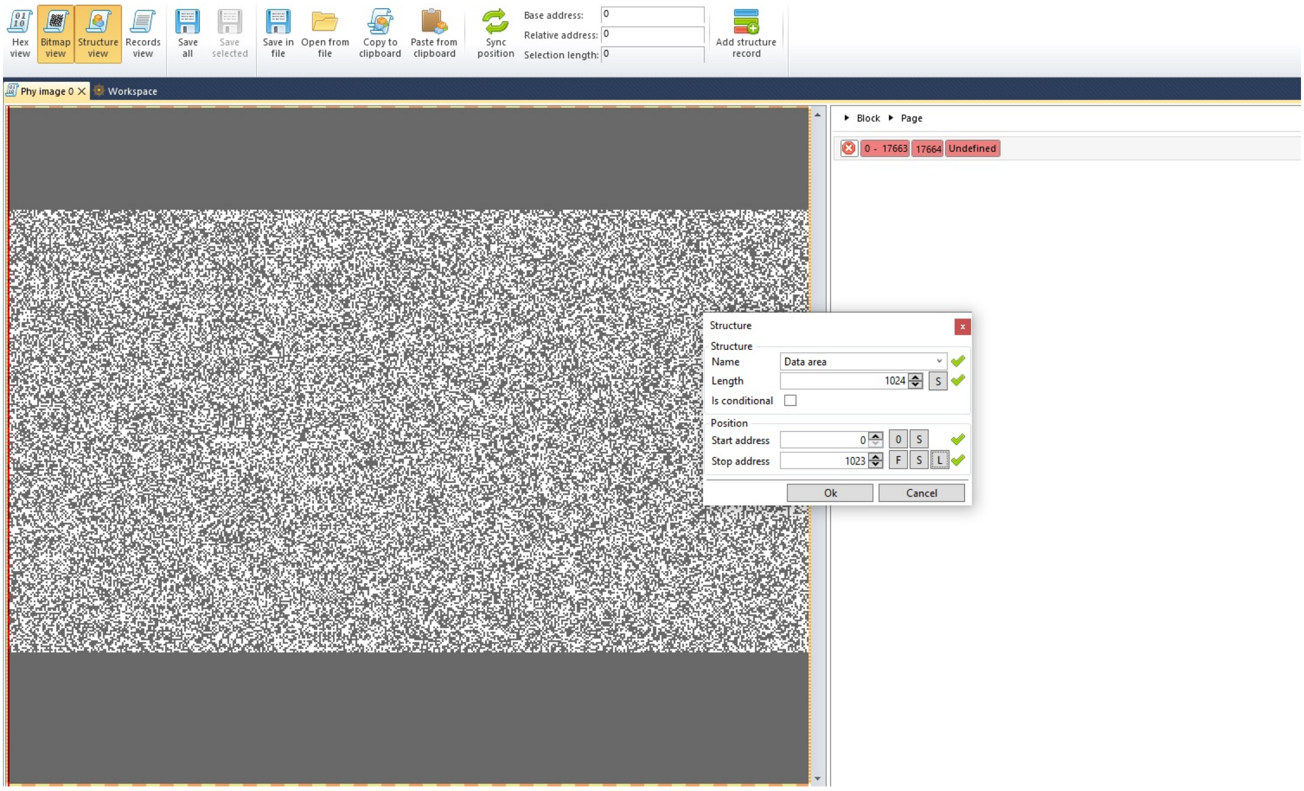
- **XORed Service area**

Not a full Service area is XORedd i n AU controllers , but just the first byte, like on the example below.



```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00008A0400 37 05 ED D1 F3 1C B2 BC 09 D8 8E 72 D9 80 0D 5E
00008A0410 35 B9 0F 4A 0E A0 A9 47 6B E7 F0 CA 43 73 B8 B8
00008A0420 E6 18 3D 46 43 77 40 41 DE E7 44 BA 5F F4 74 E7
00008A0430 2C 05 64 DD B8 0C 83 8F C4 66 B6 E3 98 79 A4 06
00008A0440 0F 0D E3 3D 77 2F 19 16 BB CF 82 83 48 0A 19 DE
00008A0450 E0 ED 70 68 BF D6 65 67 8F AA 7C C0 96 24 A8 6E
00008A0460 17 1D 8D C9 8A C9 8C 1F A9 6D CE 56 0F 5A 72 ED
00008A0470 CD AF 7B 02 14 B4 95 FB 3F B7 CF A5 8A 15 77 2B
00008A0480 3D F1 F1 06 8E 9D 44 42 E4 FF 50 6F B3 9D 4B 6E
00008A0490 86 6F 4F 17 27 7A 0A 69 FC 92 3D AA 0B EE D4 83
00008A04A0 45 69 A3 33 C1 16 E1 B2 11 36 46 76 38 86 F0 C4
00008A04B0 D5 03 8F E9 4E 85 43 9A 9E A0 A0 09 D4 E3 49 0C
00008A04C0 10 01 53 E7 D6 3A 64 6D C4 F9 BB 0B 30 60 45 6D
00008A04D0 B9 FB ED BD C0 04 C3 87 59 1D 13 D0 31 7F 53 50
00008A04E0 2C C8 AC 69 FF D7 59 9B B1 85 8E 6B 33 79 F3 C3
00008A04F0 95 72 CA D8 60 96 66 BE 55 A9 F3 F6 2D 7D 0E 5E
00008A0500 FA E3 5B 3F E0 F7 3A F7 5D 99 DC 9F A3 1D DA 4B
00008A0510 C5 CB F1 6A 0A 77 BE 88 1D FE 50 84 60 7D 80 8E
00008A0520 3C E5 B0 D0 F2 EF B5 5E F9 16 55 CB A6 49 0E 3D
00008A0530 A7 1F 74 8F D1 0C 7C 4E 0A 00 11 73 A5 ED 06 3B
00008A0540 CD 09 4D 32 B8 C9 F1 DC 97 DE CF CD F5 90 3F F0
00008A0550 52 04 2F 91 73 7E 1E 8B 4C 03 DB FF 5A 0B EC 29
00008A0560 D4 5D FE 7E 6A C0 75 BB 9C 48 4D 9F 4B C6 6B FE
00008A0570 5D E4 37 BE CA 85 52 E1 94 DC 3F 5C B8 4E 1B B6
00008A0580 AF F8 EB 6A E2 B7 AF C1 C9 F4 75 5C E3 FF A6 4D
00008A0590 9E 22 2F B7 E3 4E 44 0A C9 FF 89 0F C1 F9 28 D6
00008A05A0 58 43 BD 97 B7 31 65 AE 85 F5 96 C2 F0 2B 91 ED
00008A05B0 69 2A 5E 75 62 A2 92 1F F7 CC 41 3C 0E 81 90 0E
00008A05C0 AE 90 88 F2 43 BC 21 7C F2 5A 57 19 AE 60 E6 1D
00008A05D0 0E 72 DB 46 C2 88 D3 A4 BE B2 A9 11 40 91 02 E9
00008A05E0 2B 87 73 29 F9 56 A9 14 92 4C DF 5F 49 9D B5 BA
00008A05F0 60 09 AB DC 05 D3 F9 8E DF B2 D4 87 D2 BB A0 CB
00008A0600 BD D8 62 8A 56 D5 30 BE 70 5A BA EA D9 2B 38 F0
00008A0610 E3 0C AF E3 01 D3 32 C8 64 73 FE 14 46 A3 85 75
00008A0620 46 B3 87 C8 D6 FC 3C A9 60 50 79 32 59 B0 D6 12
00008A0630 D5 FC 32 DC 99 76 2C CF 4C EC 8D 79 5D 7F 90 06
00008A0640 9E DE 25 B7 51 20 3F E7 60 29 85 EE DC 20 05 02
00008A0650 DE 06 1A 14 AF 25 4C A2 CF D1 80 8A F1 32 B3 73
00008A0660 65 A0 89 C6 3C 11 1D A1 FD A7 C6 30 18 03 45 1B
00008A0670 7E 19 5E 18 67 2C D9 A0 34 A0 8B 44 67 CA 06 3F
00008A0680 94 CD 51 3B 47 3B E4 FC B9 D5 9C AD 8F 80 CD 47
00008A0690 B9 96 4B 58 E8 C1 B1 FC 07 EE D1 06 3F 4F 9F 3C
00008A06A0 5E A9 0F F6 87 A4 F0 0F DF 9E 7F EF 83 9D 60 38
00008A06B0 E6 DD C1 8B FF 52 10 CA 16 E2 AC 3C E9 22 1B D3
00008A06C0 A5 98 2B AE EC D0 5C 4B 12 8E 7C 33 F4 4C 3A 0E
00008A06D0 52 0E 2F BF B4 F7 B1 14 FA D7 C0 8B 4E B6 A3 63
00008A06E0 AF EB BF 0F 5E 15 1F A6 A4 B4 31 98 97 DE 3B 07
00008A06F0 E7 EC 9D F6 E7 B2 CF CC AF 65 6C FA EC D1 F5 5E
00008A0700 D0 67 3C 83 94 88 6E 29 D5 E0 80 46 BB F3 52 59
00008A0710 F0 D9 9E 30 D9 9C 57 02 C8 CD D5 23 D8 96 64 25
00008A0720 21 09 58 C2 41 62 E4 05 5E D9 0F 86 C2 BA 08 C5
00008A0730 1F B3 61 CE A8 36 C2 BC 42 E8 58 C4 51 4B 2D F5
00008A0740 50 6A 08 67 F0 FA DE A0 80 30 45 5B 39 1B 41 E3
00008A0750 A5 F0 83 F1 FD 6C 1C 8D EE CF C6 78 CB 38 05 FE
```

To assign page layout, open Dump viewer, then Structure view, and set page structures.

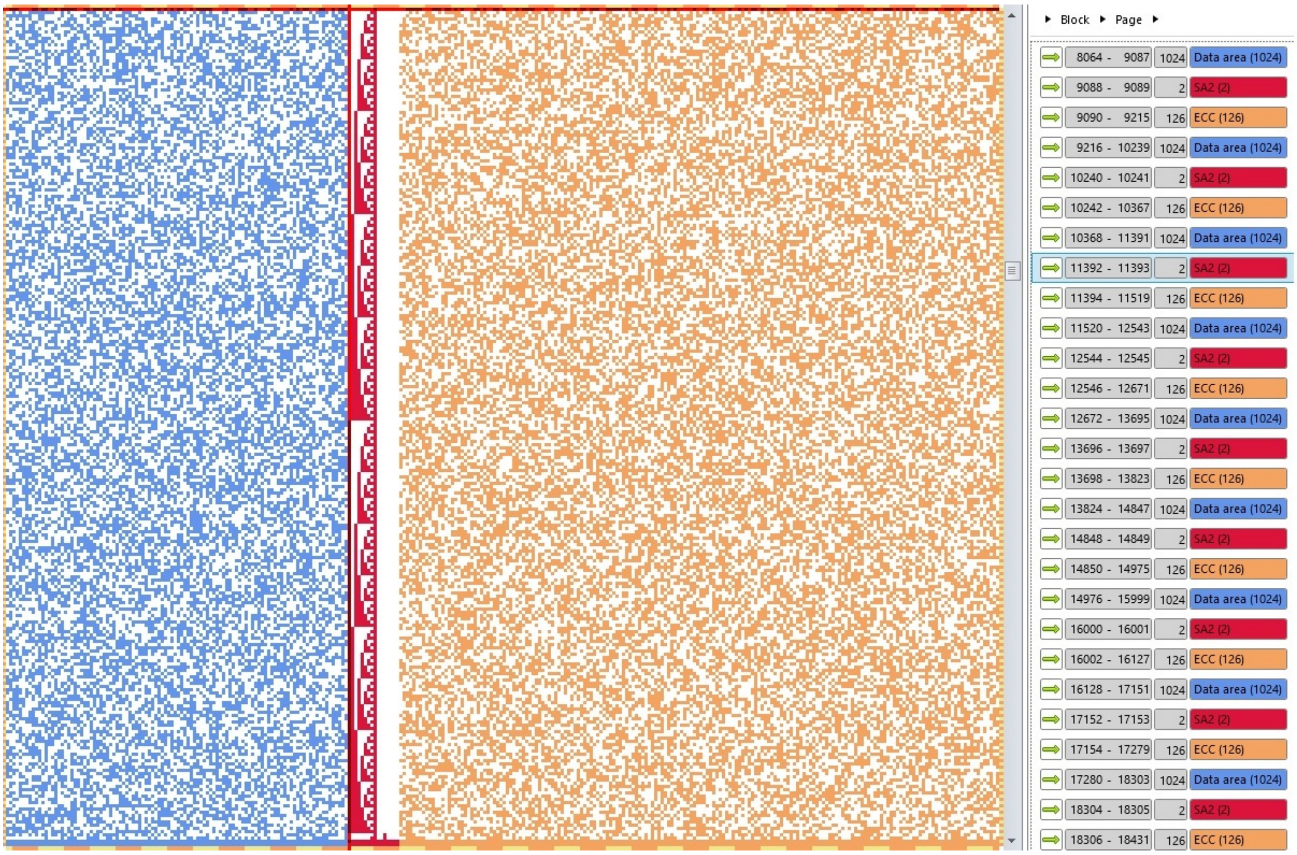


Base address: 0
Relative address: 0
Selection length: 0

0 - 17663 | 17664 | Undefined

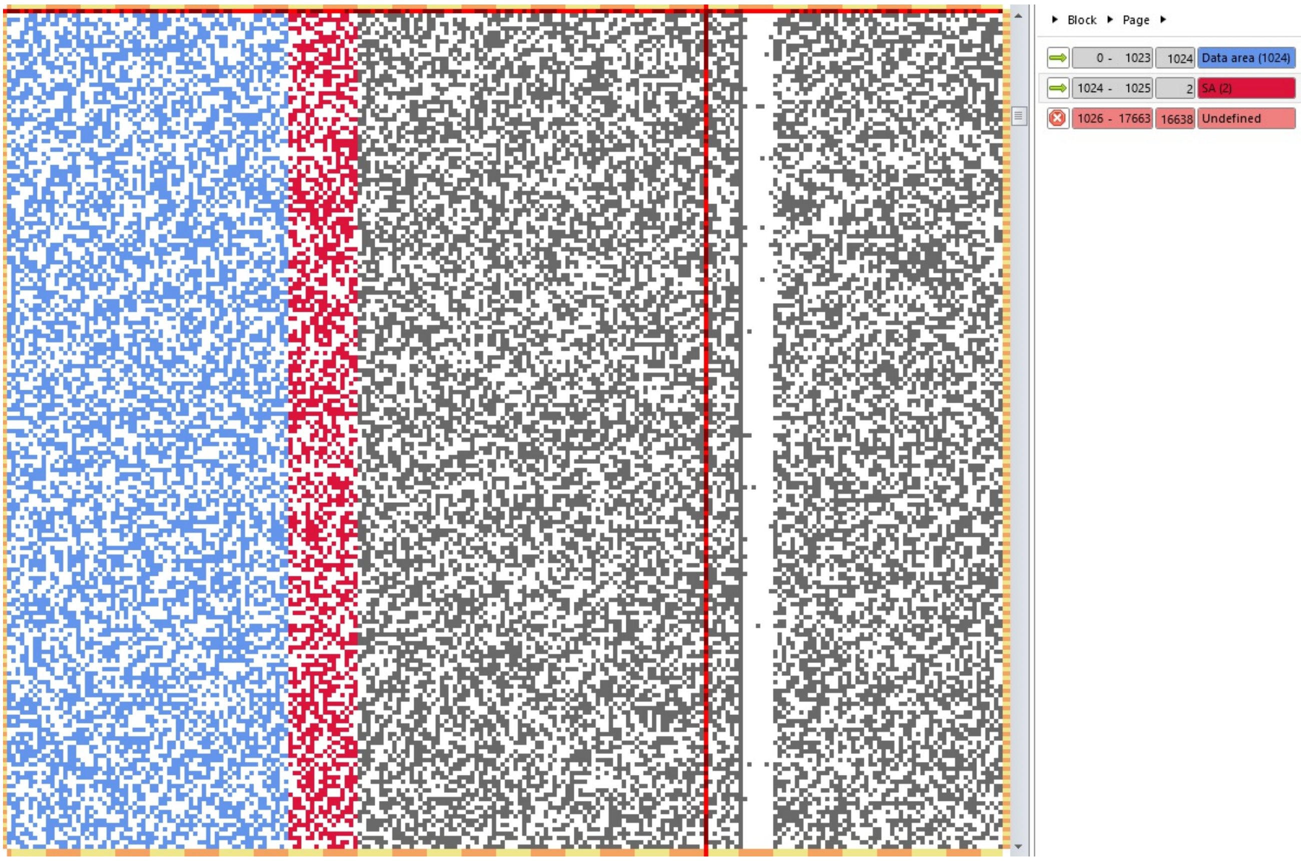
Structure
Name: Data area
Length: 1024
Is conditional ☐
Position
Start address: 0
Stop address: 1023

When page layout has been set and it corresponds to the page structures in the dump, the next step should be ECC and XOR key.



⇨	8064 - 9087	1024	Data area (1024)
⇨	9088 - 9089	2	SA2 (2)
⇨	9090 - 9215	126	ECC (126)
⇨	9216 - 10239	1024	Data area (1024)
⇨	10240 - 10241	2	SA2 (2)
⇨	10242 - 10367	126	ECC (126)
⇨	10368 - 11391	1024	Data area (1024)
⇨	11392 - 11393	2	SA2 (2)
⇨	11394 - 11519	126	ECC (126)
⇨	11520 - 12543	1024	Data area (1024)
⇨	12544 - 12545	2	SA2 (2)
⇨	12546 - 12671	126	ECC (126)
⇨	12672 - 13695	1024	Data area (1024)
⇨	13696 - 13697	2	SA2 (2)
⇨	13698 - 13823	126	ECC (126)
⇨	13824 - 14847	1024	Data area (1024)
⇨	14848 - 14849	2	SA2 (2)
⇨	14850 - 14975	126	ECC (126)
⇨	14976 - 15999	1024	Data area (1024)
⇨	16000 - 16001	2	SA2 (2)
⇨	16002 - 16127	126	ECC (126)
⇨	16128 - 17151	1024	Data area (1024)
⇨	17152 - 17153	2	SA2 (2)
⇨	17154 - 17279	126	ECC (126)
⇨	17280 - 18303	1024	Data area (1024)
⇨	18304 - 18305	2	SA2 (2)
⇨	18306 - 18431	126	ECC (126)

In the situation when set page layout doesn't match with the page structures, like on the example below, it means that there are bad columns in the dump and it will be necessary to remove them.

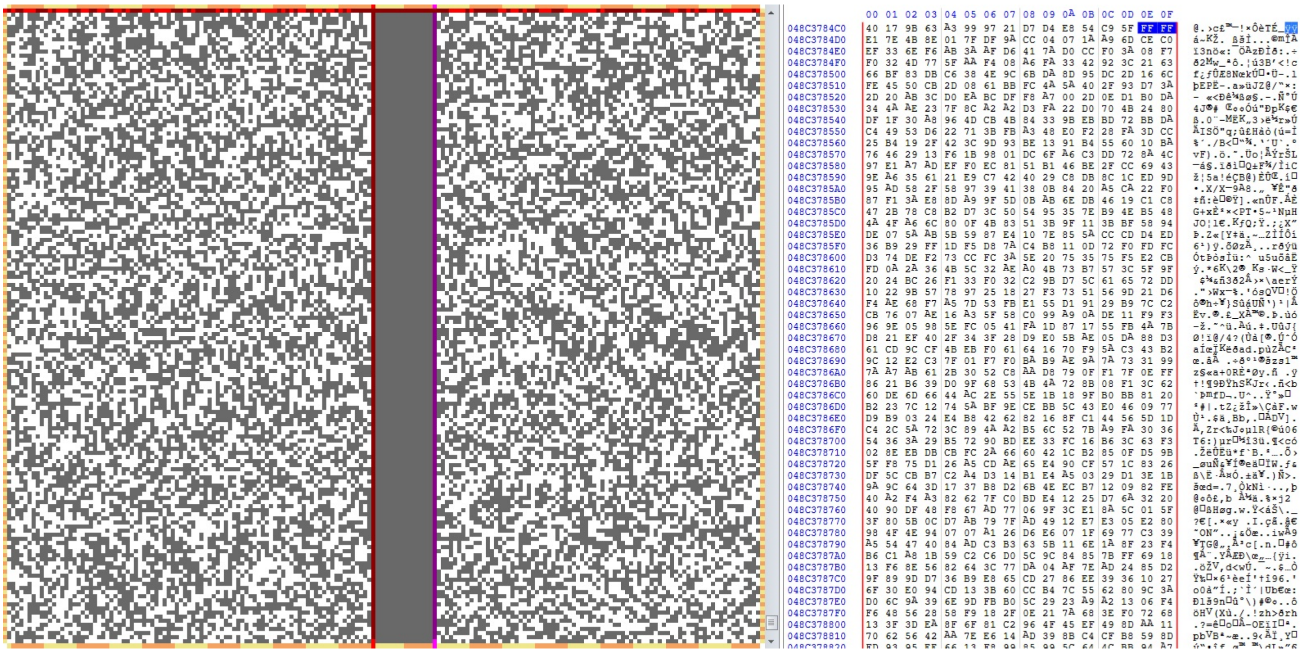


⇨	0 - 1023	1024	Data area (1024)
⇨	1024 - 1025	2	SA (2)
⊗	1026 - 17663	16638	Undefined

Bad Columns

We can distinguish two types of bad column patterns in Alcor Micro controllers, there are: 0xFF and XORed ones.

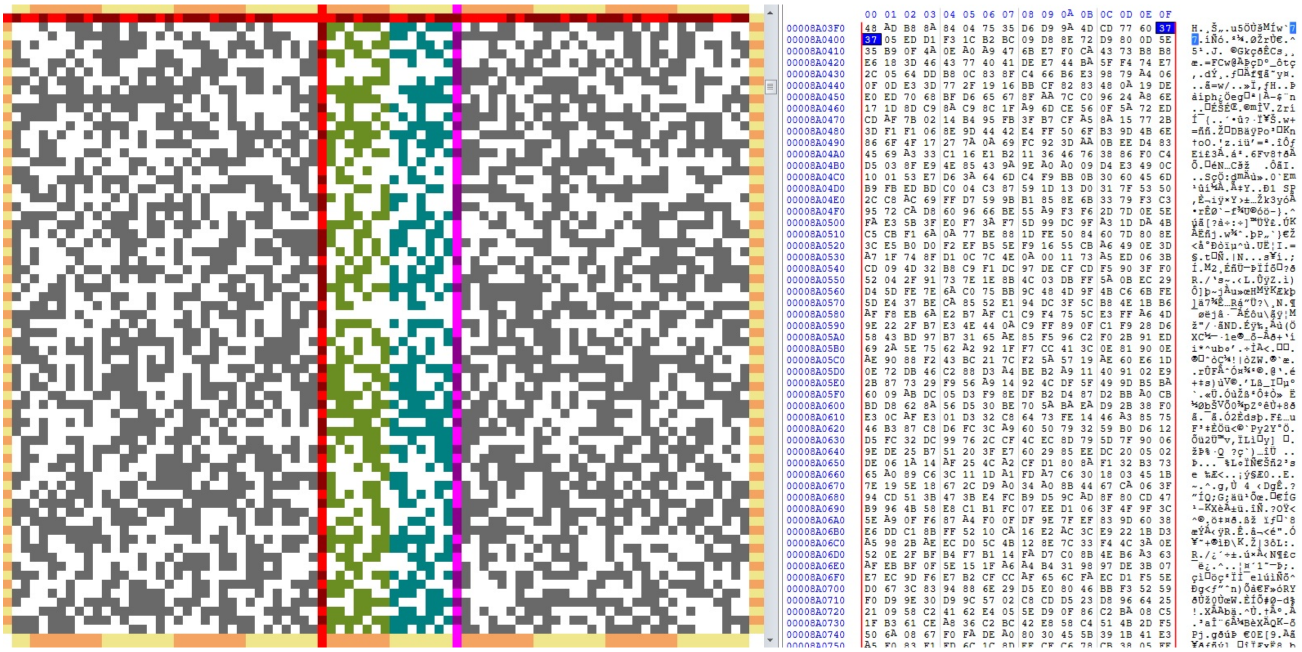
- **0xFF Bad columns**



```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
048C3784C0  40 17 9B 63 A3 99 97 21 D7 D4 E8 54 C9 5F FF FF   @.›c£™—!×ÔèTÉ_ÿÿ
048C3784D0  E1 7E 4B 8E 01 7F DF 9A CC 04 07 1A A9 6D CE C0   á~KŽ. ßšÌ...©mÎÀ
048C3784E0  EF 33 6E F6 AB 3A AF D6 41 7A D0 CC F0 3A 08 F7   ï3nö«:¯ÖAzÐÌð:.÷
048C3784F0  F0 32 4D 77 5F AA F4 08 A6 FA 33 42 92 3C 21 63   ð2Mw_ªô.¦ú3B’<!c
048C378500  66 BF 83 DB C6 38 4E 9C 6B DA 8D 95 DC 2D 16 6C   f¿ƒÛÆ8NœkÚ□•Ü-.l
048C378510  FE 45 50 CB 2D 08 61 BB FC 4A 5A 40 2F 93 D7 3A   þEPË-.a»üJZ@/“×:
048C378520  2D 20 AB 3C D0 EA BC DF F8 A7 00 2D 0E D1 B0 DA   - «<Ðê¼ßø§.-.Ñ°Ú
048C378530  34 4A AE 23 7F 8C A2 A2 D3 FA 22 D0 70 4B 24 80   4J®# Œ¢¢Óú"ÐpK$€
048C378540  DF 1F 30 A8 96 4D CB 4B 84 33 9B EB BD 72 BB DA   ß.0¨–MËK„3›ë½r»Ú
048C378550  C4 49 53 D6 22 71 3B FB A3 48 E0 F2 28 FA 3D CC   ÄISÖ"q;û£Hàò(ú=Ì
048C378560  25 B4 19 2F 42 3C 9D 93 BE 13 91 B4 55 60 10 BA   %´./B<□“¾.‘´U`.º
048C378570  76 46 29 13 F6 1B 98 01 DC 6F A6 C3 DD 72 8A 4C   vF).ö.˜.Üo¦ÃÝrŠL
048C378580  97 E1 A7 AD EF F0 EC 81 51 B1 46 BE 2F CC 69 43   —á§.ïðì□Q±F¾/ÌiC
048C378590  9E A6 35 61 21 E9 C7 42 40 29 C8 DB 8C 1C ED 9D   ž¦5a!éÇB@)ÈÛŒ.í□
048C3785A0  95 AD 58 2F 58 97 39 41 38 0B 84 20 A5 CA 22 F0   •.X/X—9A8.„ ¥Ê"ð
048C3785B0  87 F1 3A E8 8D A9 9F 5D 0B AB 6E DB 46 19 C1 C8   ‡ñ:è□©Ÿ].«nÛF.ÁÈ
048C3785C0  47 2B 78 C8 B2 D7 3C 50 54 95 35 7E B9 4E B5 48   G+xÈ²×<PT•5~¹NµH
048C3785D0  4A 4F A6 6C 80 0F 4B 83 51 3B 9F 11 3B BF 58 94   JO¦l€.KƒQ;Ÿ.;¿X”
048C3785E0  DE 07 5A AB 5B 59 87 E4 10 7E 85 5A CC CD D4 ED   Þ.Z«[Y‡ä.~…ZÌÍÔí
048C3785F0  36 B9 29 FF 1D F5 D8 7A C4 B8 11 0D 72 F0 FD FC   6¹)ÿ.õØzÄ¸..rðýü
048C378600  D3 74 DE F2 73 CC FC 3A 5E 20 75 35 75 F5 E2 CB   ÓtÞòsÌü:^ u5uõâË
048C378610  FD 0A 2A 36 4B 5C 32 AE A0 4B 73 B7 57 3C 5F 9F   ý.*6K\2® Ks·W<_Ÿ
048C378620  20 24 BC 26 F1 33 F0 32 C2 9B D7 5C 61 65 72 DD    $¼&ñ3ð2Â›×\aerÝ
048C378630  10 22 9B 57 78 97 25 18 27 F3 73 51 56 9D 21 D6   ."›Wx—%.'ósQV□!Ö
048C378640  F4 AE 68 F7 A5 7D 53 FB E1 55 D1 91 29 B9 7C C2   ô®h÷¥}SûáUÑ‘)¹|Â
048C378650  CB 76 07 AE 16 A3 5F 58 C0 99 A9 0A DE 11 F9 F3   Ëv.®.£_XÀ™©.Þ.ùó
048C378660  96 9E 05 98 5E FC 05 41 FA 1D 87 17 55 FB 4A 7B   –ž.˜^ü.Aú.‡.UûJ{
048C378670  D8 21 EF 40 2F 34 3F 28 D9 E0 5B AE 05 DA 88 D3   Ø!ï@/4?(Ùà[®.ÚˆÓ
048C378680  61 CD 9C CF 4B EB F0 61 64 16 70 F9 5A C3 43 B2   aÍœÏKëðad.pùZÃC²
048C378690  9C 12 E2 C3 7F 01 F7 F0 BA B9 AE 9A 7A 73 31 99   œ.âÃ .÷ðº¹®šzs1™
048C3786A0  7A A7 AB 61 2B 30 52 C8 AA D8 79 0F F1 7F 0E FF   z§«a+0RÈªØy.ñ .ÿ
048C3786B0  86 21 B6 39 D0 9F 68 53 4B 4A 72 8B 08 F1 3C 62   †!¶9ÐŸhSKJr‹.ñ<b
048C3786C0  60 DE 6D 66 44 AC 2E 55 5E 1B 18 9F B0 BB 81 20   `ÞmfD¬.U^..Ÿ°»□
048C3786D0  B2 23 7C 12 74 5A BF 9E CE BB 5C 43 E0 46 09 77   ²#|.tZ¿žÎ»\CàF.w
048C3786E0  D9 B9 03 24 E4 B8 42 62 82 16 8F C1 44 56 5D 1D   Ù¹.$ä¸Bb‚.□ÁDV].
048C3786F0  C4 2C 5A 72 3C 89 4A A2 B5 6C 52 7B A9 FA 30 36   Ä,Zr<‰J¢µlR{©ú06
048C378700  54 36 3A 29 B5 72 90 BD EE 33 FC 16 B6 3C 63 F3   T6:)µr□½î3ü.¶<có
048C378710  02 8E EB DB CB FC 2A 66 60 42 1C B2 85 0F D5 9B   .ŽëÛËü*f`B.²….Õ›
048C378720  5F F8 75 D1 26 A5 CD AE 65 E4 90 CF 57 1C 83 26   _øuÑ&¥Í®eä□ÏW.ƒ&
048C378730  DF 5C CB B7 C2 A4 D3 14 B1 E4 A5 03 29 D1 3E 1B   ß\Ë·Â¤Ó.±ä¥.)Ñ>.
048C378740  9A 9C 64 3D 17 37 B8 D2 6B 4E EC B7 12 09 82 FE   šœd=.7¸ÒkNì·..‚þ
048C378750  40 A2 F4 A3 82 62 7F C0 BD E4 12 25 D7 6A 32 20   @¢ô£‚b À½ä.%×j2 
048C378760  40 90 DF 48 F8 67 AD 77 06 9F 3C E1 8A 5C 01 5F   @□ßHøg.w.Ÿ<áŠ\._
048C378770  3F 80 5B 0C D7 AB 79 7F AD 49 12 E7 E3 05 E2 80   ?€[.×«y .I.çã.â€
048C378780  98 4F 4E 94 07 07 A1 26 D6 E6 07 1F 69 77 C3 39   ˜ON”..¡&Öæ..iwÃ9
048C378790  A5 54 47 40 84 AD C3 B3 63 5B 11 6E 1A 8F 23 F4   ¥TG@„.Ã³c[.n.□#ô
048C3787A0  B6 C1 A8 1B 59 C2 C6 D0 5C 9C 84 85 7B FF 69 18   ¶Á¨.YÂÆÐ\œ„…{ÿi.
048C3787B0  13 F6 8E 56 82 64 3C 77 DA 04 AF 7E AD 24 85 D2   .öŽV‚d<wÚ.¯~.$…Ò
048C3787C0  9F 89 9D D7 36 B9 E8 65 CD 27 86 EE 39 36 10 27   Ÿ‰□×6¹èeÍ'†î96.'
048C3787D0  6F 30 E0 94 CD 13 3B 60 CC B4 7C 55 62 80 9C 3A   o0à”Í.;`Ì´|Ub€œ:
048C3787E0  D0 6C 9A 39 6E 9D FB B0 5C 29 23 A9 A2 13 06 F4   Ðlš9n□û°\)#©¢..ô
048C3787F0  F6 48 56 28 58 F9 18 2F 0E 21 7A 68 3E F0 72 68   öHV(Xù./.!zh>ðrh
048C378800  13 3F 3D EA 8F 6F 81 C2 96 4F 45 EF 49 8D AA 11   .?=ê□o□Â–OEïI□ª.
048C378810  70 62 56 42 AA 7E E6 14 AD 39 8B C4 CF B8 59 8D   pbVBª~æ..9‹ÄÏ¸Y□
```

The standard type of bad column pattern is common for the majority of NAND controllers.

- **XORed bad columns**



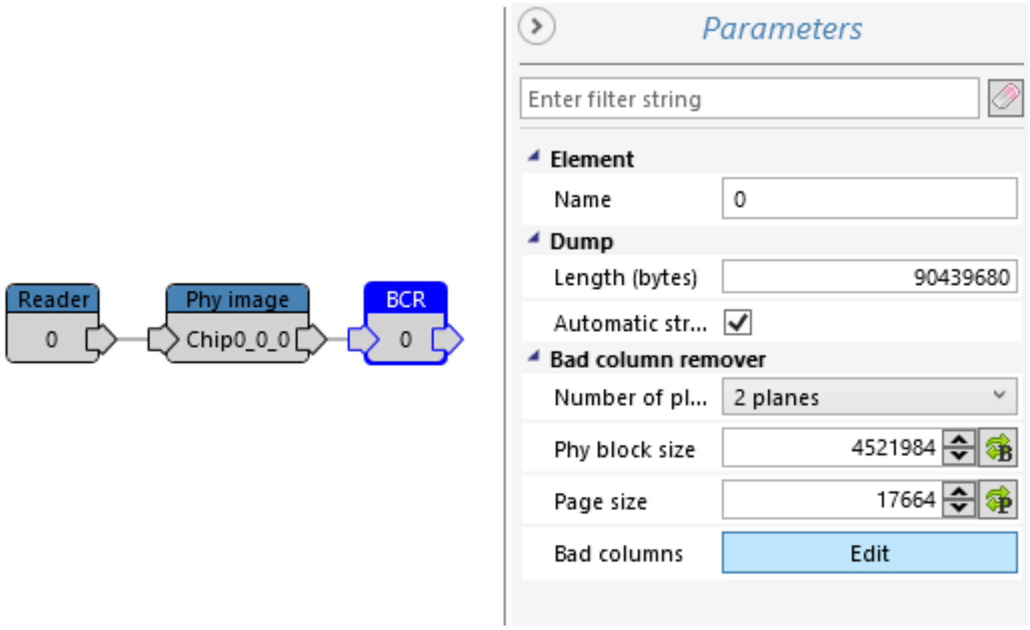
```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00008A03F0 48 AD B8 8A 84 04 75 35 D6 D9 9A 4D CD 77 60 37  H.¸Š„.u5ÖÙšMÍw`7
00008A0400 37 05 ED D1 F3 1C B2 BC 09 D8 8E 72 D9 80 0D 5E  7.íÑó.²¼.ØŽrÙ€.^
00008A0410 35 B9 0F 4A 0E A0 A9 47 6B E7 F0 CA 43 73 B8 B8  5¹.J. ©GkçðÊCs¸¸
00008A0420 E6 18 3D 46 43 77 40 41 DE E7 44 BA 5F F4 74 E7  æ.=FCw@AÞçDº_ôtç
00008A0430 2C 05 64 DD B8 0C 83 8F C4 66 B6 E3 98 79 A4 06  ,.dÝ¸.ƒ□Äf¶ã˜y¤.
00008A0440 0F 0D E3 3D 77 2F 19 16 BB CF 82 83 48 0A 19 DE  ..ã=w/..»Ï‚ƒH..Þ
00008A0450 E0 ED 70 68 BF D6 65 67 8F AA 7C C0 96 24 A8 6E  àíph¿Öeg□ª|À–$¨n
00008A0460 17 1D 8D C9 8A C9 8C 1F A9 6D CE 56 0F 5A 72 ED  ..□ÉŠÉŒ.©mÎV.Zrí
00008A0470 CD AF 7B 02 14 B4 95 FB 3F B7 CF A5 8A 15 77 2B  Í¯{..´•û?·Ï¥Š.w+
00008A0480 3D F1 F1 06 8E 9D 44 42 E4 FF 50 6F B3 9D 4B 6E  =ññ.Ž□DBäÿPo³□Kn
00008A0490 86 6F 4F 17 27 7A 0A 69 FC 92 3D AA 0B EE D4 83  †oO.'z.iü’=ª.îÔƒ
00008A04A0 45 69 A3 33 C1 16 E1 B2 11 36 46 76 38 86 F0 C4  Ei£3Á.á².6Fv8†ðÄ
00008A04B0 D5 03 8F E9 4E 85 43 9A 9E A0 A0 09 D4 E3 49 0C  Õ.□éN…Cšž  .ÔãI.
00008A04C0 10 01 53 E7 D6 3A 64 6D C4 F9 BB 0B 30 60 45 6D  ..SçÖ:dmÄù».0`Em
00008A04D0 B9 FB ED BD C0 04 C3 87 59 1D 13 D0 31 7F 53 50  ¹ûí½À.Ã‡Y..Ð1 SP
00008A04E0 2C C8 AC 69 FF D7 59 9B B1 85 8E 6B 33 79 F3 C3  ,È¬iÿ×Y›±…Žk3yóÃ
00008A04F0 95 72 CA D8 60 96 66 BE 55 A9 F3 F6 2D 7D 0E 5E  •rÊØ`–f¾U©óö-}.^
00008A0500 FA E3 5B 3F E0 F7 3A F7 5D 99 DC 9F A3 1D DA 4B  úã[?à÷:÷]™ÜŸ£.ÚK
00008A0510 C5 CB F1 6A 0A 77 BE 88 1D FE 50 84 60 7D 80 8E  ÅËñj.w¾ˆ.þP„`}€Ž
00008A0520 3C E5 B0 D0 F2 EF B5 5E F9 16 55 CB A6 49 0E 3D  <å°ÐòïµˆùUË¦I.=
00008A0530 A7 1F 74 8F D1 0C 7C 4E 0A 00 11 73 A5 ED 06 3B  §.t□Ñ.|N...s¥í.;
00008A0540 CD 09 4D 32 B8 C9 F1 DC 97 DE CF CD F5 90 3F F0  Í.M2¸ÉñÜ—ÞÏÍõ□?ð
00008A0550 52 04 2F 91 73 7E 1E 8B 4C 03 DB FF 5A 0B EC 29  R./‘s~.‹L.ÛÿZ.ì)
00008A0560 D4 5D FE 7E 6A C0 75 BB 9C 48 4D 9F 4B C6 6B FE  Ô]þ~jÀu»œHMŸKÆkþ
00008A0570 5D E4 37 BE CA 85 52 E1 94 DC 3F 5C B8 4E 1B B6  ]ä7¾Ê…Rá”Ü?\¸N.¶
00008A0580 AF F8 EB 6A E2 B7 AF C1 C9 F4 75 5C E3 FF A6 4D  ¯øëjâ·¯ÁÉôu\ãÿ¦M
00008A0590 9E 22 2F B7 E3 4E 44 0A C9 FF 89 0F C1 F9 28 D6  ž"/·ãND.Éÿ‰.Áù(Ö
00008A05A0 58 43 BD 97 B7 31 65 AE 85 F5 96 C2 F0 2B 91 ED  XC½—·1e®…õ–Âð+‘í
00008A05B0 69 2A 5E 75 62 A2 92 1F F7 CC 41 3C 0E 81 90 0E  i*^ub¢’.÷ÌA<.□□.
00008A05C0 AE 90 88 F2 43 BC 21 7C F2 5A 57 19 AE 60 E6 1D  ®□ˆòC¼!|òZW.®`æ.
00008A05D0 0E 72 DB 46 C2 88 D3 A4 BE B2 A9 11 40 91 02 E9  .rÛFÂˆÓ¤¾²©.@‘.é
00008A05E0 2B 87 73 29 F9 56 A9 14 92 4C DF 5F 49 9D B5 BA  +‡s)ùV©.’LßI□µº
00008A05F0 60 09 AB DC 05 D3 F9 8E DF B2 D4 87 D2 BB A0 CB  `.«Ü.ÓùŽß²Ô‡Ò» Ë
00008A0600 BD D8 62 8A 56 D5 30 BE 70 5A BA EA D9 2B 38 F0  ½ØbŠVÕ0¾pZºêÙ+8ð
00008A0610 E3 0C AF E3 01 D3 32 C8 64 73 FE 14 46 A3 85 75  ã.¯ã.Ó2Èdsþ.F£…u
00008A0620 46 B3 87 C8 D6 FC 3C A9 60 50 79 32 59 B0 D6 12  F³‡ÈÖü<©`Py2Y°Ö.
00008A0630 D5 FC 32 DC 99 76 2C CF 4C EC 8D 79 5D 7F 90 06  Õü2Ü™v,ÏLì□y] □.
00008A0640 9E DE 25 B7 51 20 3F E7 60 29 85 EE DC 20 05 02  žÞ%·Q ?ç`)…îÜ ..
00008A0650 DE 06 1A 14 AF 25 4C A2 CF D1 80 8A F1 32 B3 73  Þ...¯%L¢ÏÑ€Šñ2³s
00008A0660 65 A0 89 C6 3C 11 1D A1 FD A7 C6 30 18 03 45 1B  e ‰Æ<..¡ý§Æ0..E.
00008A0670 7E 19 5E 18 67 2C D9 A0 34 A0 8B 44 67 CA 06 3F  ~.^.g,Ù 4 ‹DgÊ.?
00008A0680 94 CD 51 3B 47 3B E4 FC B9 D5 9C AD 8F 80 CD 47  ”ÍQ;G;äü¹Õœ.□€ÍG
00008A0690 B9 96 4B 58 E8 C1 B1 FC 07 EE D1 06 3F 4F 9F 3C  ¹–KXèÁ±ü.îÑ.?OŸ<
00008A06A0 5E A9 0F F6 87 A4 F0 0F DF 9E 7F EF 83 9D 60 38  ^©.ö‡¤ð.ßž ïƒ□`8
00008A06B0 E6 DD C1 8B FF 52 10 CA 16 E2 AC 3C E9 22 1B D3  æÝÁ‹ÿR.Ê.â¬<é".Ó
00008A06C0 A5 98 2B AE EC D0 5C 4B 12 8E 7C 33 F4 4C 3A 0E  ¥˜+®ìÐ\K.Ž|3ôL:.
00008A06D0 52 0E 2F BF B4 F7 B1 14 FA D7 C0 8B 4E B6 A3 63  R./¿´÷±.ú×À‹N¶£c
00008A06E0 AF EB BF 0F 5E 15 1F A6 A4 B4 31 98 97 DE 3B 07  ¯ë¿.^..¦¤´1˜—Þ;.
00008A06F0 E7 EC 9D F6 E7 B2 CF CC AF 65 6C FA EC D1 F5 5E  çì□öç²ÏÌ¯elúìÑõ^
00008A0700 D0 67 3C 83 94 88 6E 29 D5 E0 80 46 BB F3 52 59  Ðg<ƒ”ˆn)Õà€F»óRY
00008A0710 F0 D9 9E 30 D9 9C 57 02 C8 CD D5 23 D8 96 64 25  ðÙž0ÙœW.ÈÍÕ#Ø–d%
00008A0720 21 09 58 C2 41 62 E4 05 5E D9 0F 86 C2 BA 08 C5  !.XÂAbä.^Ù.†Âº.Å
00008A0730 1F B3 61 CE A8 36 C2 BC 42 E8 58 C4 51 4B 2D F5  .³aÎ¨6Â¼BèXÄQK-õ
00008A0740 50 6A 08 67 F0 FA DE A0 80 30 45 5B 39 1B 41 E3  Pj.gðúÞ €0E[9.Aã
```

This type of bad columns is very hard to notice since these columns don't have the same patterns, unlike 0xFF's, but their pattern depends on the previous byte of data. We assume that those controllers write bytes, which they already have in the buffer, onto bad column position. The picture above presents an example of such a column, both neighbor columns have the same patterns and values in hex. The first one is a user data and the second (duplicated one), is a bad column and it will be necessary to remove it.

Removal of XORed bad columns

To remove bad columns from the dump attach BadColumnRemover (BCR) element to the Physical image element and click Edit.

> ⓘ The detailed description of this tool is available in the video: "NEW automated Bad Column Remover. Case studies".


Reader
0
Phy image
Chip0_0_0
BCR
0
Parameters
Enter filter string
Element
Dump
Bad column remover

Name: 0

Length (bytes): 90439680

Automatic str...: ☑

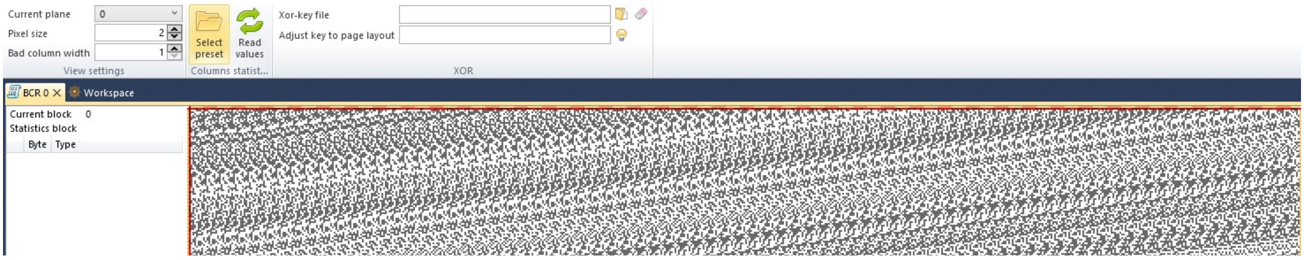
Number of pl...: 2 planes

Phy block size: 4521984

Page size: 17664

Bad columns: Edit

In the first step, in the BCR menu, it's necessary to attach a special preset with a set of rules for Alcor Micro XORed bad columns. To do that click "Select preset" and select "AlcorMicro_AUxxxx_Xored".


Select preset
Read values
View settings
Columns statist...
Xor-key file
Adjust key to page layout
XOR
BCR 0
Workspace
Statistics block
Byte
Type

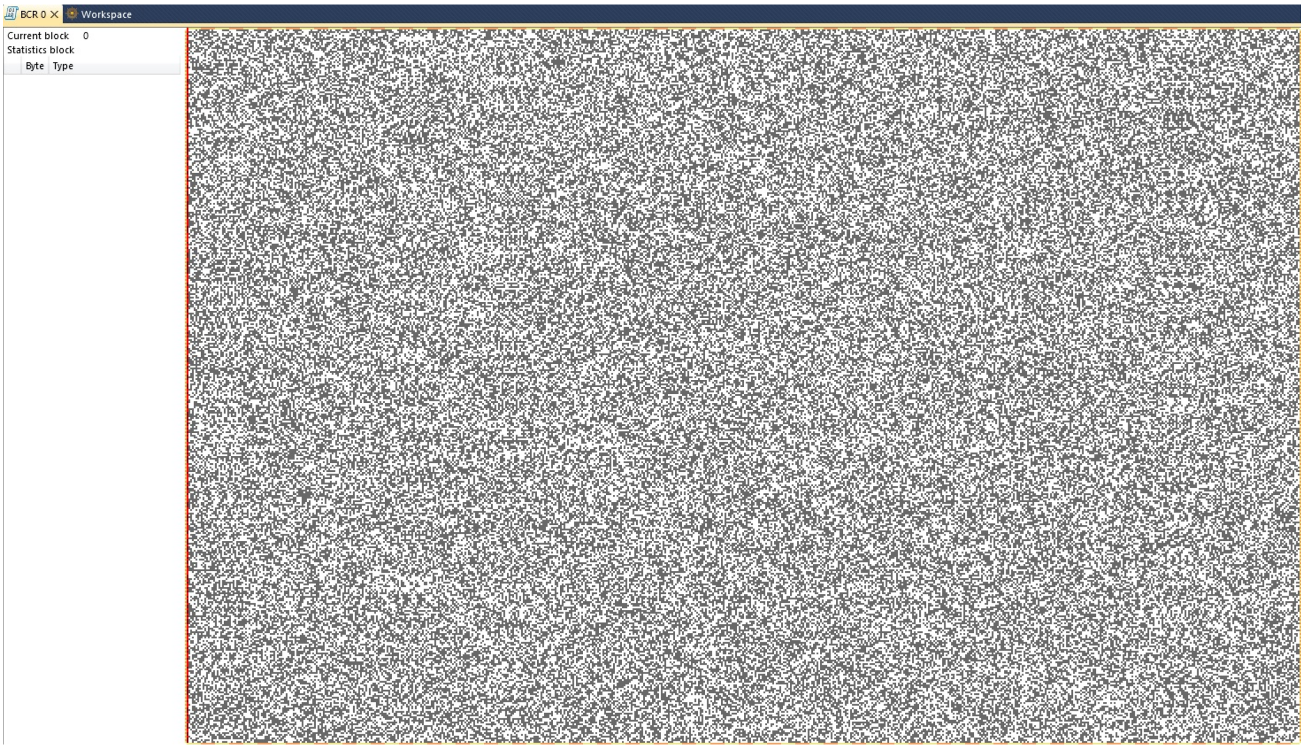
Current plane: 0

Pixel size: 2

Bad column width: 1

Current block: 0

Now you will need to find a block with user data (noise).

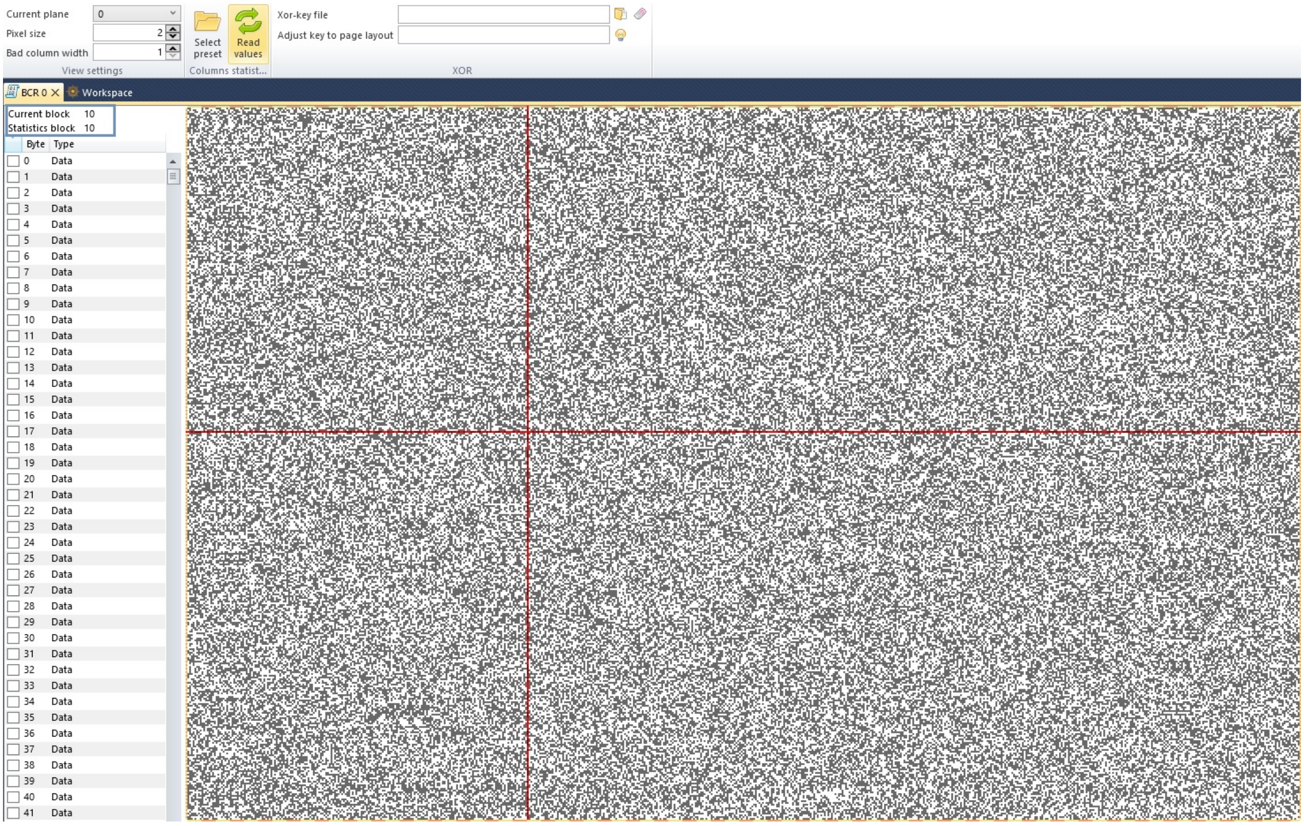

BCR 0
Workspace
Statistics block
Byte
Type

Current block: 0

The list of NAND binary patterns, in bitmap viewer, is available here: https://support.rusolut.com/portal/en/kb/articles/binary-patterns-in-nand-flash-memory-12-3-2020

> ⓘ It is important to select block with user data, because reading statistics from blocks that contain the XOR key may cause false positives results since binary statistics of them are very similar on each iteration. Therefore these blocks should be skipped.

When a proper block has been found you will need to select it with the Left Mouse Button and click "Read values", to detect the type of each column (byte). The numbers of the currently selected block and the analyzed block are located in the panel on the left side (Current block, Statistics block).

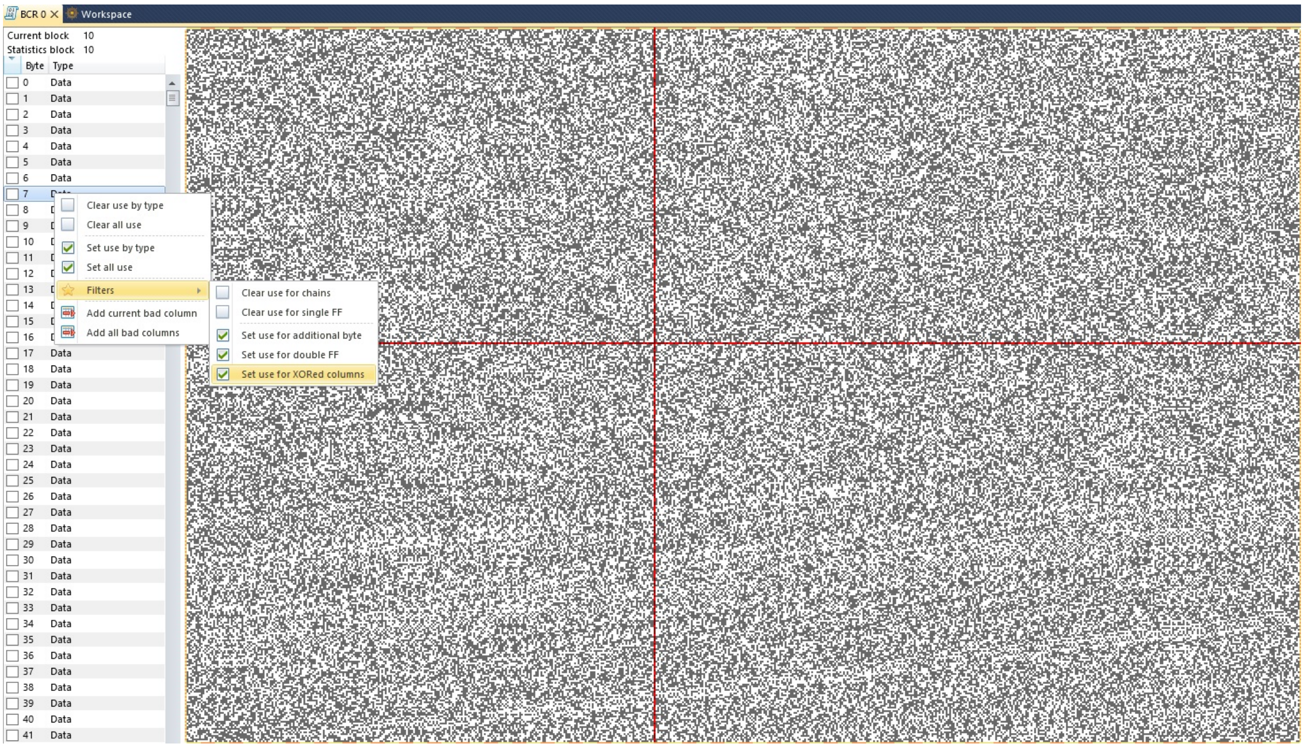


Current plane 0
Pixel size 2
Bad column width 1
Xor-key file
Adjust key to page layout

Current block 10
Statistics block 10

	Byte	Type
☐	0	Data
☐	1	Data
☐	2	Data
☐	3	Data
☐	4	Data
☐	5	Data
☐	6	Data
☐	7	Data
☐	8	Data
☐	9	Data
☐	10	Data
☐	11	Data
☐	12	Data
☐	13	Data
☐	14	Data
☐	15	Data
☐	16	Data
☐	17	Data
☐	18	Data
☐	19	Data
☐	20	Data
☐	21	Data
☐	22	Data
☐	23	Data
☐	24	Data
☐	25	Data
☐	26	Data
☐	27	Data
☐	28	Data
☐	29	Data
☐	30	Data
☐	31	Data
☐	32	Data
☐	33	Data
☐	34	Data
☐	35	Data
☐	36	Data
☐	37	Data
☐	38	Data
☐	39	Data
☐	40	Data
☐	41	Data

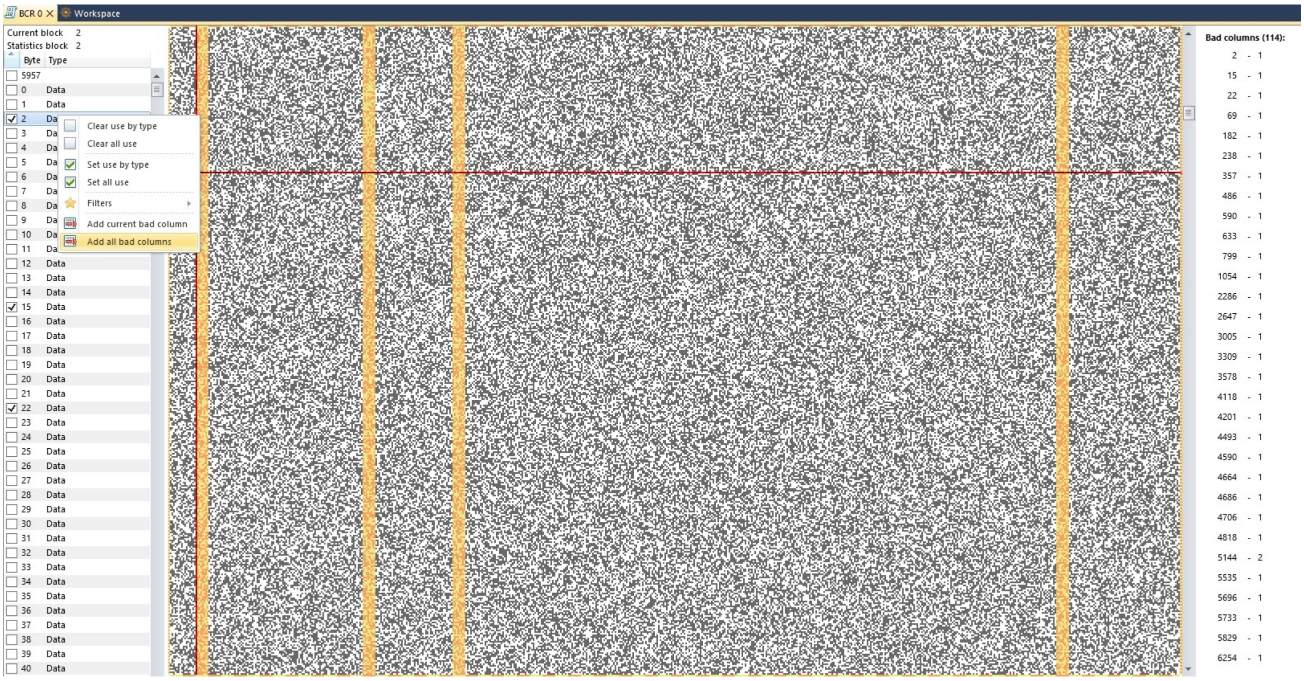
To locate the positions of XORed bad columns click Right Mouse Button on a random Byte Index and chose option "Set use for XORed columns".


BCR 0
Workspace
Clear use by type
Clear all use
Set use by type
Set all use
Filters
Add current bad column
Add all bad columns
Clear use for chains
Clear use for single FF
Set use for additional byte
Set use for double FF
Set use for XORed columns

```
Current block     10
Statistics block  10
```

Byte	Type
☐ 0	Data
☐ 1	Data
☐ 2	Data
☐ 3	Data
☐ 4	Data
☐ 5	Data
☐ 6	Data
☐ 7	Data
☐ 8	D
☐ 9	D
☐ 10	D
☐ 11	D
☐ 12	D
☐ 13	D
☐ 14	D
☐ 15	D
☐ 16	D
☐ 17	Data
☐ 18	Data
☐ 19	Data
☐ 20	Data
☐ 21	Data
☐ 22	Data
☐ 23	Data
☐ 24	Data
☐ 25	Data
☐ 26	Data
☐ 27	Data
☐ 28	Data
☐ 29	Data
☐ 30	Data
☐ 31	Data
☐ 32	Data
☐ 33	Data
☐ 34	Data
☐ 35	Data
☐ 36	Data
☐ 37	Data
☐ 38	Data
☐ 39	Data
☐ 40	Data
☐ 41	Data

Bad columns have been detected, now it's possible to add them by clicking "Add all bad columns".

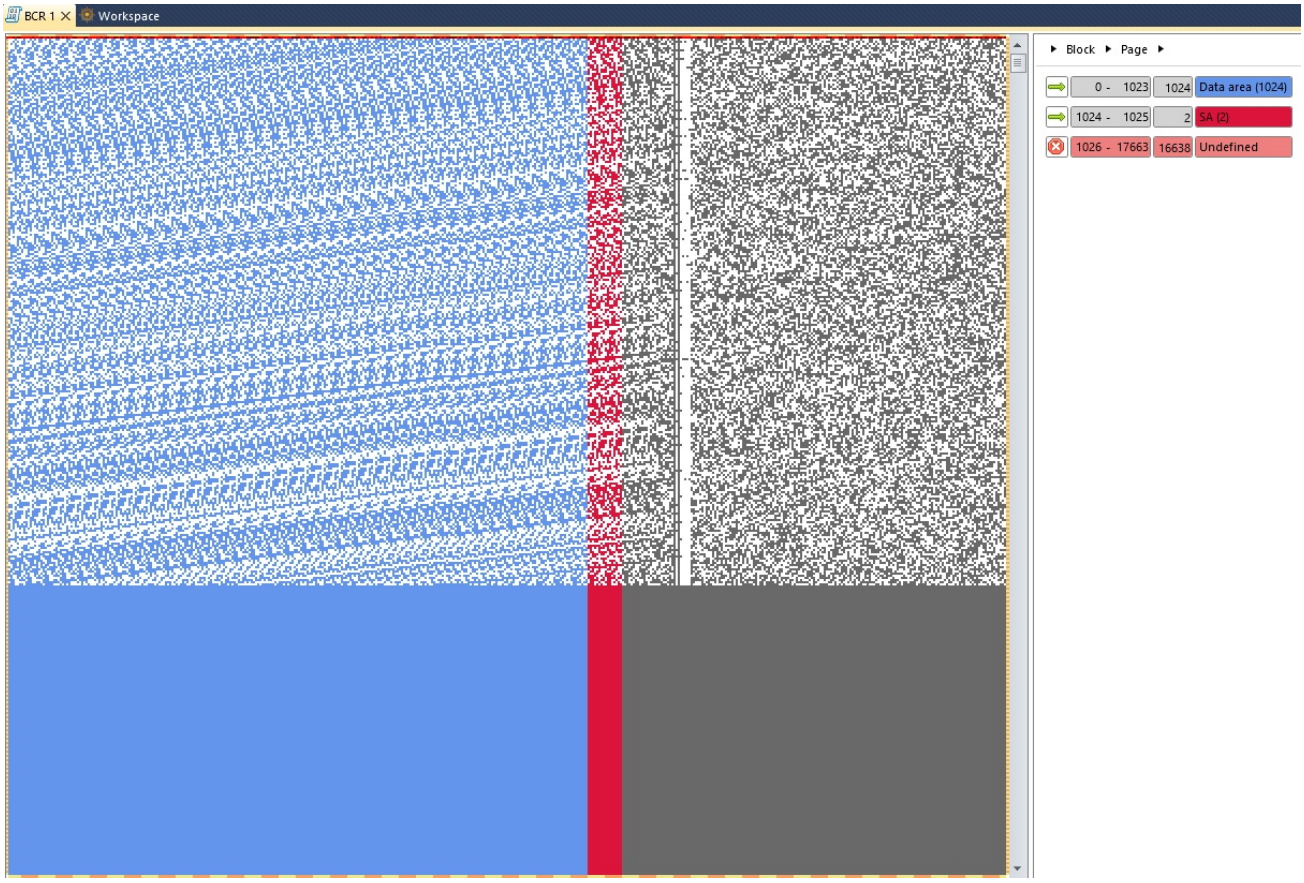

BCR 0
Workspace
Clear use by type
Clear all use
Set use by type
Set all use
Filters
Add current bad column
Add all bad columns

```
Current block     2
Statistics block  2
```

Byte	Type
☐ 5957	
☐ 0	Data
☐ 1	Data
☑ 2	Da
☐ 3	Da
☐ 4	Da
☐ 5	Da
☐ 6	Da
☐ 7	Da
☐ 8	Da
☐ 9	Da
☐ 10	Da
☐ 11	Da
☐ 12	Data
☐ 13	Data
☐ 14	Data
☑ 15	Data
☐ 16	Data
☐ 17	Data
☐ 18	Data
☐ 19	Data
☐ 20	Data
☐ 21	Data
☑ 22	Data
☐ 23	Data
☐ 24	Data
☐ 25	Data
☐ 26	Data
☐ 27	Data
☐ 28	Data
☐ 29	Data
☐ 30	Data
☐ 31	Data
☐ 32	Data
☐ 33	Data
☐ 34	Data
☐ 35	Data
☐ 36	Data
☐ 37	Data
☐ 38	Data
☐ 39	Data
☐ 40	Data

```
Bad columns (114):
   2 - 1
  15 - 1
  22 - 1
  69 - 1
 182 - 1
 238 - 1
 357 - 1
 486 - 1
 590 - 1
 633 - 1
 799 - 1
1054 - 1
2286 - 1
2647 - 1
3005 - 1
3309 - 1
3578 - 1
4118 - 1
4201 - 1
4493 - 1
4590 - 1
4664 - 1
4686 - 1
4706 - 1
4818 - 1
5144 - 2
5535 - 1
5696 - 1
5733 - 1
5829 - 1
6254 - 1
```

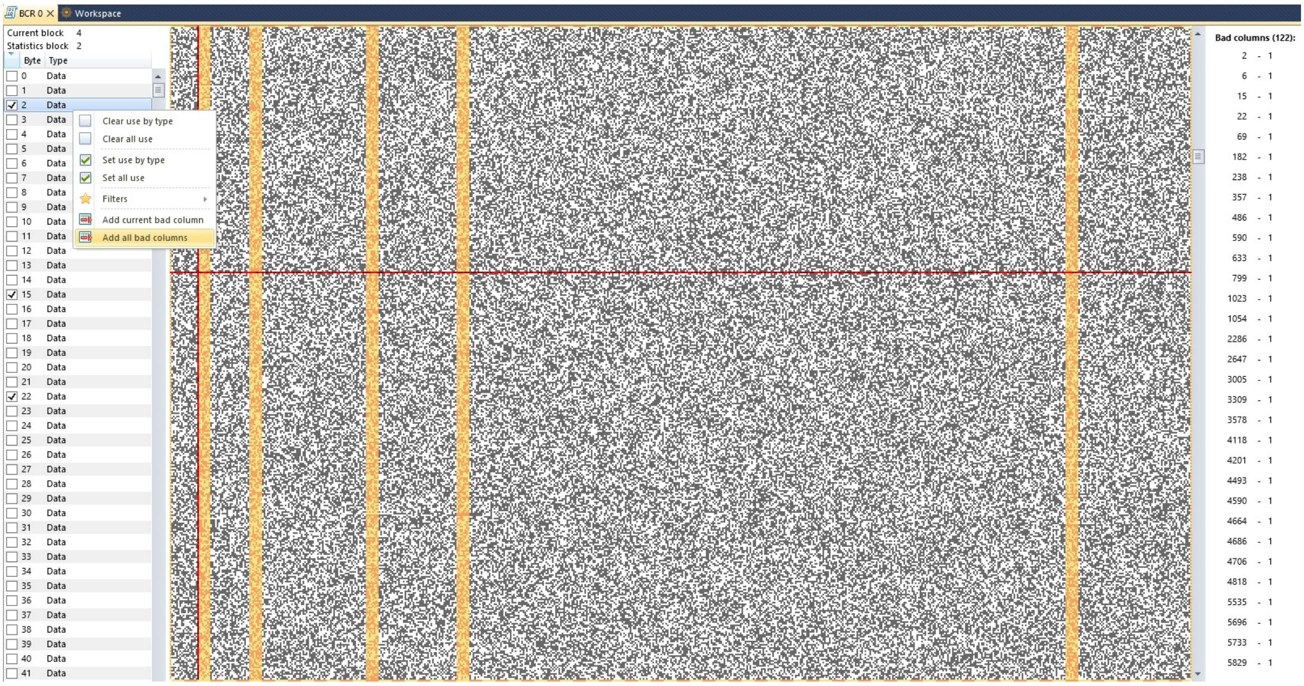
In the next step, to check if all bad columns have been removed, you will need to open the BCR element and check if the set page layout corresponds to the page structures in the dump.



0 - 1023	1024	Data area (1024)
1024 - 1025	2	SA (2)
1026 - 17663	16638	Undefined

Page structure doesn't match, it means that bad columns are remaining in the dump. Such a situation may appear especially when there are a lot of bit errors in the dump, in these cases, it is necessary to repeat the procedure of bad column removal several times and analyze different blocks with data.

Open Bad Column Remover, attach preset, and read values from the different blocks with data. Then "Set use for XORed columns" and "Add all bad columns". Previously, after analysis of a single block, 114 Bad columns were detected, after analysis of 3 another blocks amount is increased to 122.

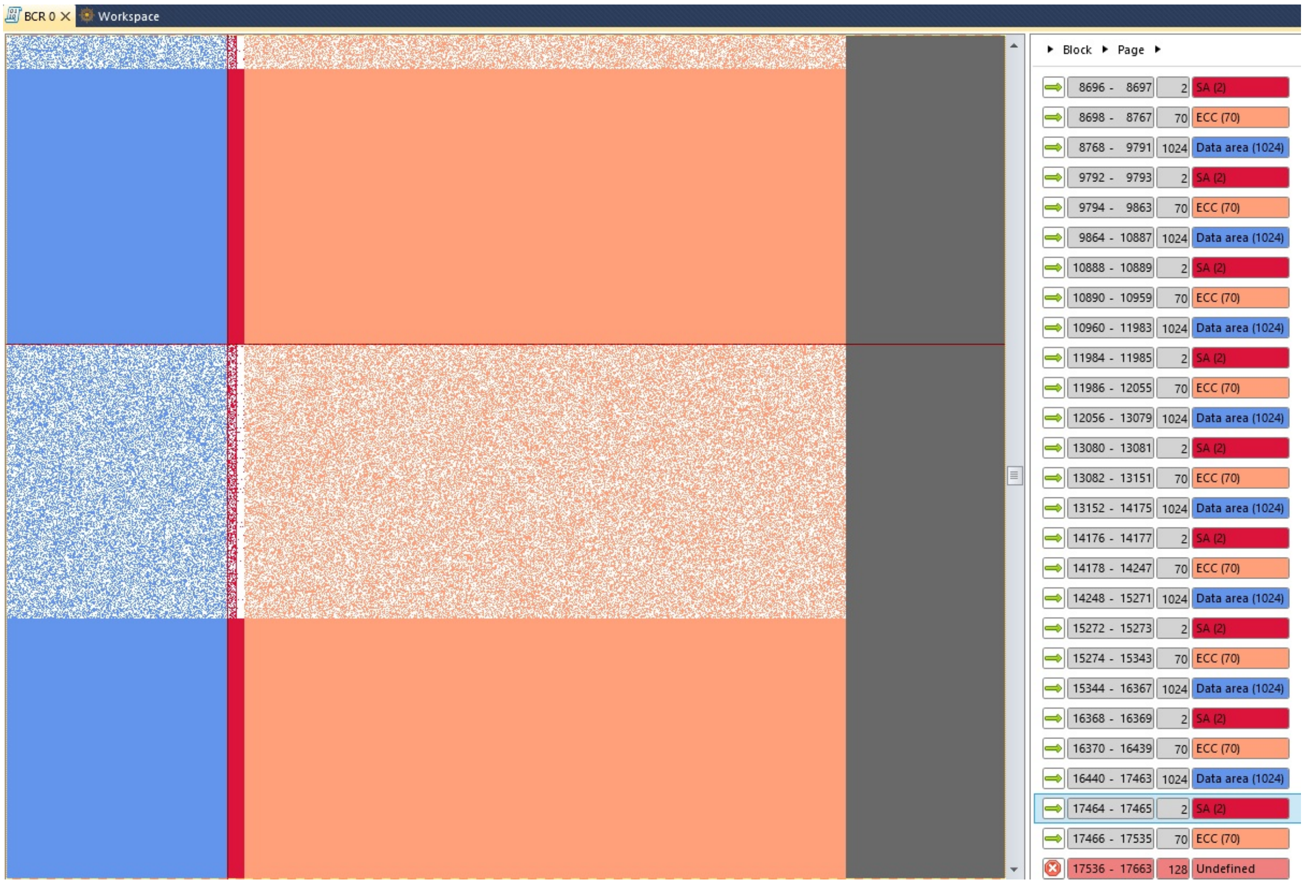


Current block 4
Statistics block 2

	Byte	Type
☐	0	Data
☐	1	Data
☑	2	Data
☐	3	Data
☐	4	Data
☐	5	Data
☐	6	Data
☐	7	Data
☐	8	Data
☐	9	Data
☐	10	Data
☐	11	Data
☐	12	Data
☐	13	Data
☐	14	Data
☑	15	Data
☐	16	Data
☐	17	Data
☐	18	Data
☐	19	Data
☐	20	Data
☐	21	Data
☑	22	Data
☐	23	Data
☐	24	Data
☐	25	Data
☐	26	Data
☐	27	Data
☐	28	Data
☐	29	Data
☐	30	Data
☐	31	Data
☐	32	Data
☐	33	Data
☐	34	Data
☐	35	Data
☐	36	Data
☐	37	Data
☐	38	Data
☐	39	Data
☐	40	Data
☐	41	Data

Bad columns (122):
2 - 1
6 - 1
15 - 1
22 - 1
69 - 1
182 - 1
238 - 1
357 - 1
486 - 1
590 - 1
633 - 1
799 - 1
1023 - 1
1054 - 1
2286 - 1
2647 - 1
3005 - 1
3309 - 1
3578 - 1
4118 - 1
4201 - 1
4493 - 1
4590 - 1
4664 - 1
4686 - 1
4706 - 1
4818 - 1
5535 - 1
5696 - 1
5733 - 1
5829 - 1

To check if all bad columns are removed page layout should be checked once again.



➡	8696 - 8697	2	SA (2)
➡	8698 - 8767	70	ECC (70)
➡	8768 - 9791	1024	Data area (1024)
➡	9792 - 9793	2	SA (2)
➡	9794 - 9863	70	ECC (70)
➡	9864 - 10887	1024	Data area (1024)
➡	10888 - 10889	2	SA (2)
➡	10890 - 10959	70	ECC (70)
➡	10960 - 11983	1024	Data area (1024)
➡	11984 - 11985	2	SA (2)
➡	11986 - 12055	70	ECC (70)
➡	12056 - 13079	1024	Data area (1024)
➡	13080 - 13081	2	SA (2)
➡	13082 - 13151	70	ECC (70)
➡	13152 - 14175	1024	Data area (1024)
➡	14176 - 14177	2	SA (2)
➡	14178 - 14247	70	ECC (70)
➡	14248 - 15271	1024	Data area (1024)
➡	15272 - 15273	2	SA (2)
➡	15274 - 15343	70	ECC (70)
➡	15344 - 16367	1024	Data area (1024)
➡	16368 - 16369	2	SA (2)
➡	16370 - 16439	70	ECC (70)
➡	16440 - 17463	1024	Data area (1024)
➡	17464 - 17465	2	SA (2)
➡	17466 - 17535	70	ECC (70)
⊗	17536 - 17663	128	Undefined

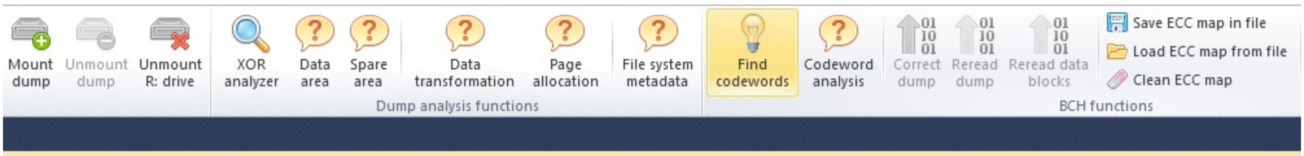
The page layout is matching correctly, so all bad columns have been removed successfully.

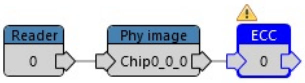
ECC

In the majority cases controllers calculate their Error Correction Codes from XORed data, such controllers are scrambling user data at first, and then start the calculation of the ECC for the previously scrambled portion of data. In those situations, we are connecting the ECC element to the scrambled dump "Physical Image" or "BCR" element, in case of bad columns. Another situation takes place when the controller has calculated ECC from raw, non-XORed data. To correct the dump in such cases it's necessary to remove XOR key influence from the dump and then find matching ECC codewords. In Alcor Micro controllers Error Codes may be calculated both these ways and it's necessary to figure out which one is correct.

> ⓘ More information about Error Correction Codes are available in the article ECC in NAND flash memory and in the video Chip reading. ECC usage. Page layout. Dump reread. Read Retry.

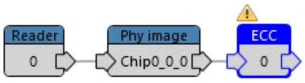
To check this, you will need to try the ECC search before and after the XOR for Data area element. Connect ECC to the source dump element and click "Find codewords".

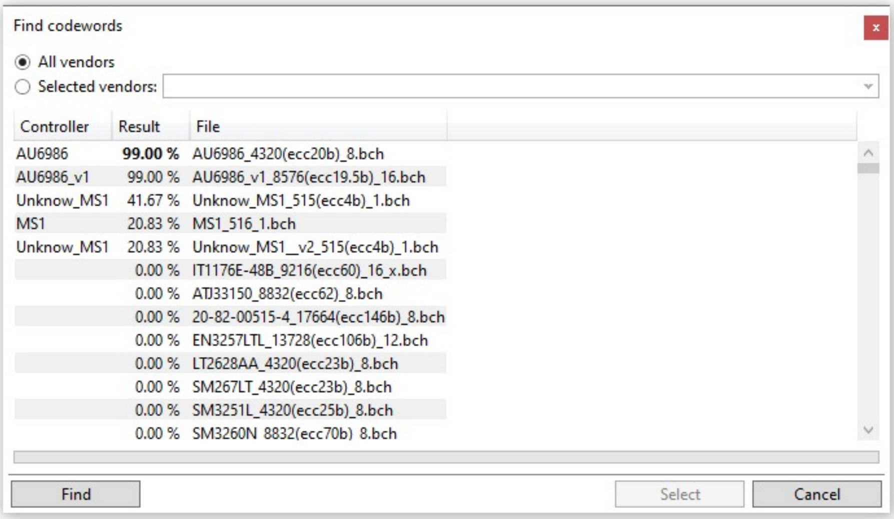

Mount dump
Unmount dump
Unmount R: drive
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions
Find codewords
Codeword analysis
Correct dump
Reread dump
Reread data blocks
Save ECC map in file
Load ECC map from file
Clean ECC map
BCH functions


Reader
0
Phy image
Chip0_0_0
ECC
0

If a matching codeword has been found, data transformation (INVERSION, XOR) should be determined in the next step .

Mount dump
Unmount dump
Unmount R: drive
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions
Find codewords
Codeword analysis
Correct dump
Reread dump
Reread data blocks
Save ECC map in file
Load ECC map from file
Clean ECC map
BCH functions


Reader
0
Phy image
Chip0_0_0
ECC
0


Find codewords

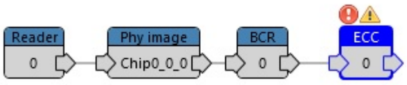
● All vendors
○ Selected vendors:

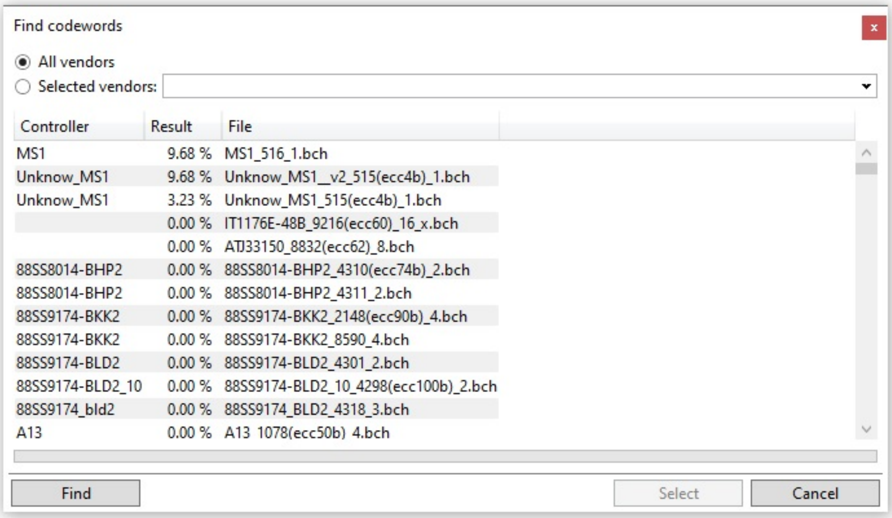
Controller	Result	File
AU6986	**99.00 %**	AU6986_4320(ecc20b)_8.bch
AU6986_v1	99.00 %	AU6986_v1_8576(ecc19.5b)_16.bch
Unknow_MS1	41.67 %	Unknow_MS1_515(ecc4b)_1.bch
MS1	20.83 %	MS1_516_1.bch
Unknow_MS1	20.83 %	Unknow_MS1__v2_515(ecc4b)_1.bch
	0.00 %	IT1176E-48B_9216(ecc60)_16_x.bch
	0.00 %	ATJ33150_8832(ecc62)_8.bch
	0.00 %	20-82-00515-4_17664(ecc146b)_8.bch
	0.00 %	EN3257LTL_13728(ecc106b)_12.bch
	0.00 %	LT2628AA_4320(ecc23b)_8.bch
	0.00 %	SM267LT_4320(ecc23b)_8.bch
	0.00 %	SM3251L_4320(ecc25b)_8.bch
	0.00 %	SM3260N 8832(ecc70b) 8.bch

Find
Select
Cancel

In the situation when ECC wasn't found it's necessary to check several additional things.

Mount dump
Unmount dump
Unmount R: drive
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions
Find codewords
Codeword analysis
Correct dump
Reread dump
Reread data blocks
Save ECC map in file
Load ECC map from file
Clean ECC map
BCH functions


Reader
0
Phy image
Chip0_0_0
BCR
0
ECC
0

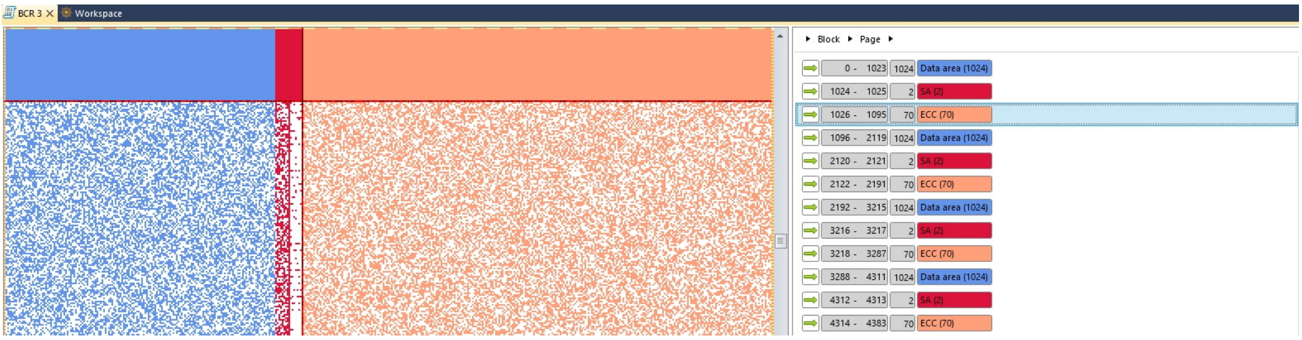

Find codewords

● All vendors
○ Selected vendors:

Controller	Result	File
MS1	9.68 %	MS1_516_1.bch
Unknow_MS1	9.68 %	Unknow_MS1__v2_515(ecc4b)_1.bch
Unknow_MS1	3.23 %	Unknow_MS1_515(ecc4b)_1.bch
	0.00 %	IT1176E-48B_9216(ecc60)_16_x.bch
	0.00 %	ATJ33150_8832(ecc62)_8.bch
88SS8014-BHP2	0.00 %	88SS8014-BHP2_4310(ecc74b)_2.bch
88SS8014-BHP2	0.00 %	88SS8014-BHP2_4311_2.bch
88SS9174-BKK2	0.00 %	88SS9174-BKK2_2148(ecc90b)_4.bch
88SS9174-BKK2	0.00 %	88SS9174-BKK2_8590_4.bch
88SS9174-BLD2	0.00 %	88SS9174-BLD2_4301_2.bch
88SS9174-BLD2_10	0.00 %	88SS9174-BLD2_10_4298(ecc100b)_2.bch
88SS9174_bld2	0.00 %	88SS9174_BLD2_4318_3.bch
A13	0.00 %	A13 1078(ecc50b) 4.bch

Find
Select
Cancel

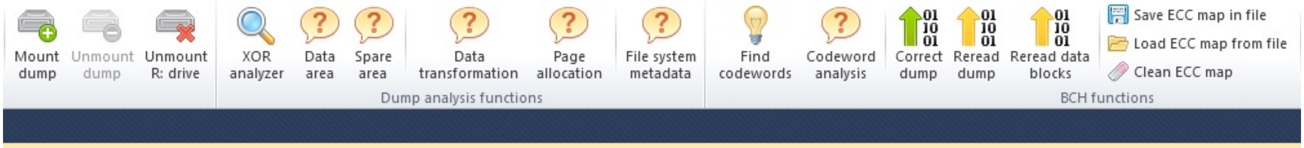
First is the size of the ECC area.

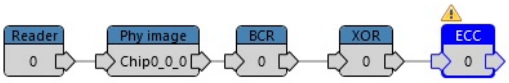

BCR 3
Workspace
Block
Page

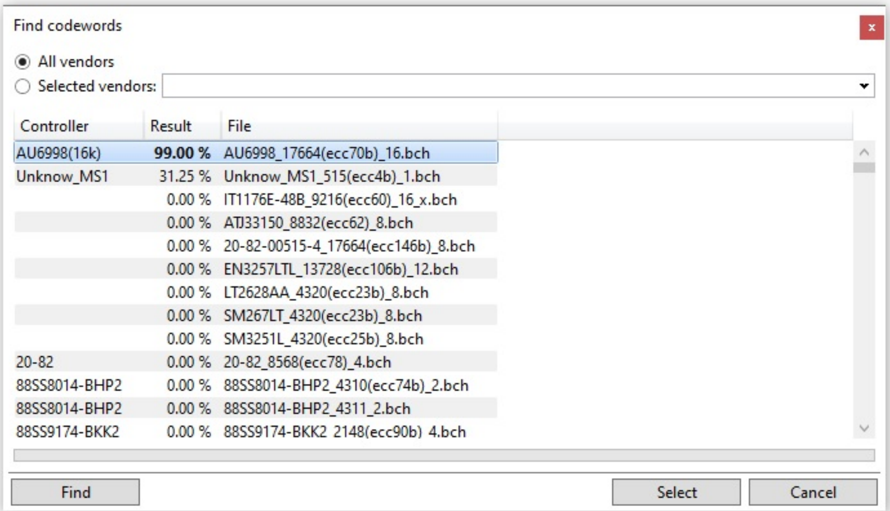
0 - 1023	1024	Data area (1024)
1024 - 1025	2	SA (2)
1026 - 1095	70	ECC (70)
1096 - 2119	1024	Data area (1024)
2120 - 2121	2	SA (2)
2122 - 2191	70	ECC (70)
2192 - 3215	1024	Data area (1024)
3216 - 3217	2	SA (2)
3218 - 3287	70	ECC (70)
3288 - 4311	1024	Data area (1024)
4312 - 4313	2	SA (2)
4314 - 4383	70	ECC (70)

If the size is supported then ECC should be tested after XOR for Data Area. Check chapter **"XOR key determination - Data area"**.

ⓘ The list of all supported ECCs for Alcor Micro controllers is available in VNR\DataBase\BCHCodewords\AlcorMicro(AU) folder.


Mount dump
Unmount dump
Unmount R: drive
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions
Find codewords
Codeword analysis
Correct dump
Reread dump
Reread data blocks
Save ECC map in file
Load ECC map from file
Clean ECC map
BCH functions


Reader
0
Phy image
Chip0_0_0
BCR
0
XOR
0
ECC
0

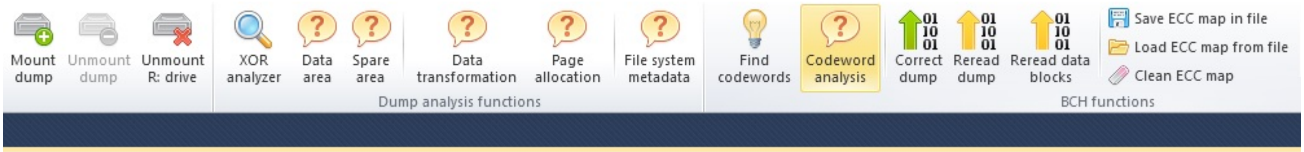

Find codewords

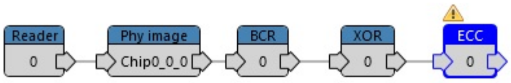
● All vendors
○ Selected vendors:

Controller	Result	File
AU6998(16k)	**99.00 %**	AU6998_17664(ecc70b)_16.bch
Unknow_MS1	31.25 %	Unknow_MS1_515(ecc4b)_1.bch
	0.00 %	IT1176E-48B_9216(ecc60)_16_x.bch
	0.00 %	ATJ33150_8832(ecc62)_8.bch
	0.00 %	20-82-00515-4_17664(ecc146b)_8.bch
	0.00 %	EN3257LTL_13728(ecc106b)_12.bch
	0.00 %	LT2628AA_4320(ecc23b)_8.bch
	0.00 %	SM267LT_4320(ecc23b)_8.bch
	0.00 %	SM3251L_4320(ecc25b)_8.bch
20-82	0.00 %	20-82_8568(ecc78)_4.bch
88SS8014-BHP2	0.00 %	88SS8014-BHP2_4310(ecc74b)_2.bch
88SS8014-BHP2	0.00 %	88SS8014-BHP2_4311_2.bch
88SS9174-BKK2	0.00 %	88SS9174-BKK2_2148(ecc90b)_4.bch

Find
Select
Cancel

In the case when the size is not supported, ECC should be tested manually with the "Codeword analysis" tool, before and after XOR for the Data area element.


Mount dump
Unmount dump
Unmount R: drive
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions
Find codewords
Codeword analysis
Correct dump
Reread dump
Reread data blocks
Save ECC map in file
Load ECC map from file
Clean ECC map
BCH functions

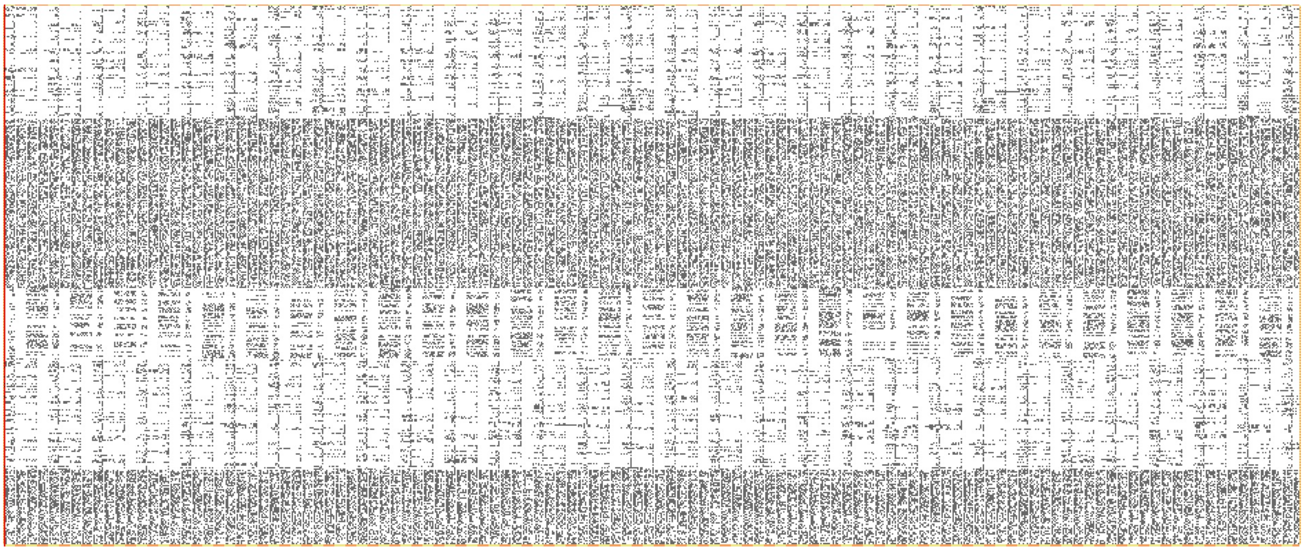

Reader
0
Phy image
Chip0_0_0
BCR
0
XOR
0
ECC
0

ⓘ Description of how to use this tool is available in the article ECC in NAND flash memory , chapter **"Unsupported ECC - how to make a new code"** .

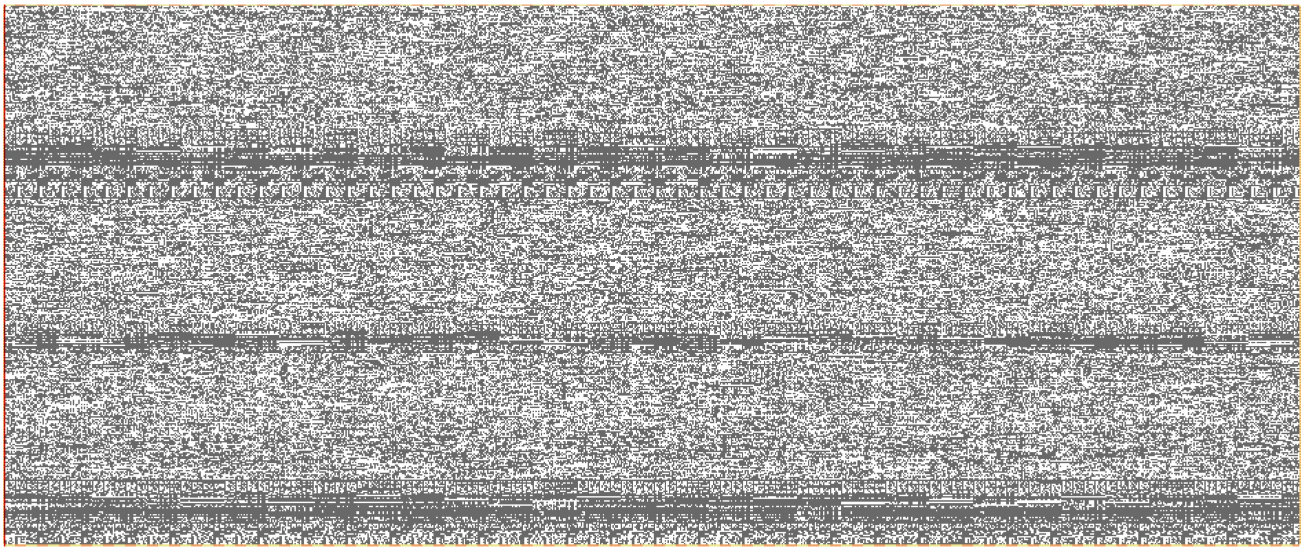
Data transformations

Controllers of NAND memories are known from their data transformations which are applying on user data. Those transformations follow from few considerations, like speeding up of input operations (INVERSION) or increase entropy inside NAND TLC chips (XOR) to keep user data more stable. In some old devices equipped with an SLC chips user data may not be transformed at all. Each of these transformations can be observed through the Bitmap Viewer.

- **No transformation** - Not transformed (raw) user data. Visually user data contain more zeros than ones.

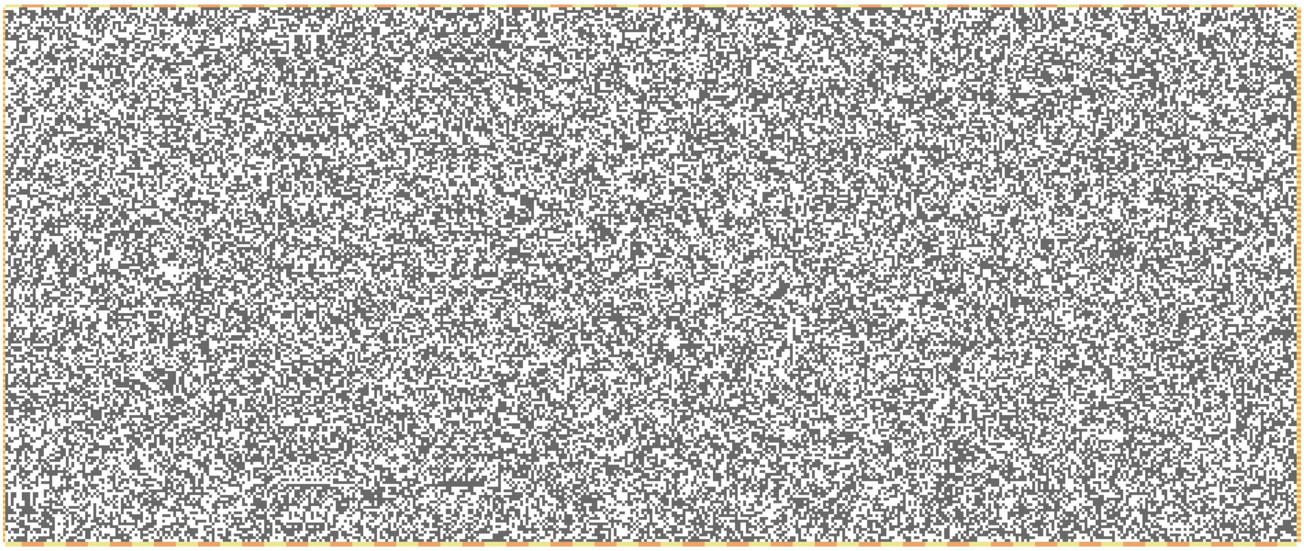


- **Inversion** - Binaries of user data, which was inverted, consist of the majority of ones (grey pixels).



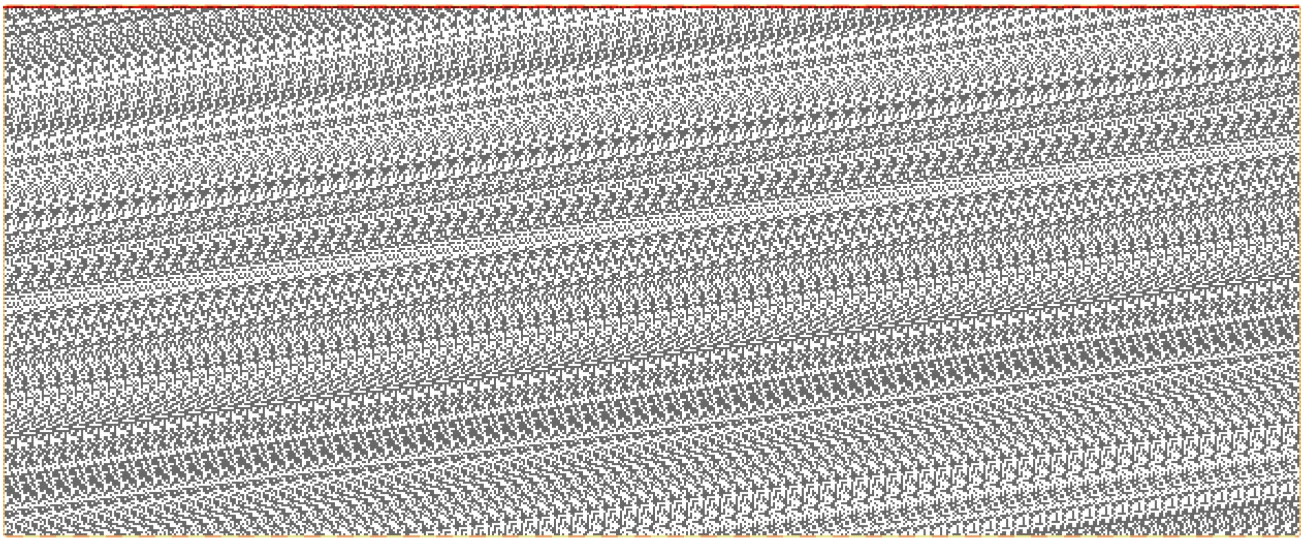
ⓘ More details about controllers and their data transformations are available in the video: Inversion. Data transformation analysis. XOR. XOR analyzer .

- **XOR** - No patterns of user data (noise) in the bitmap viewer.



In Alcor Micro cases we are using two types of XOR keys, for the Data area and for the Service area.

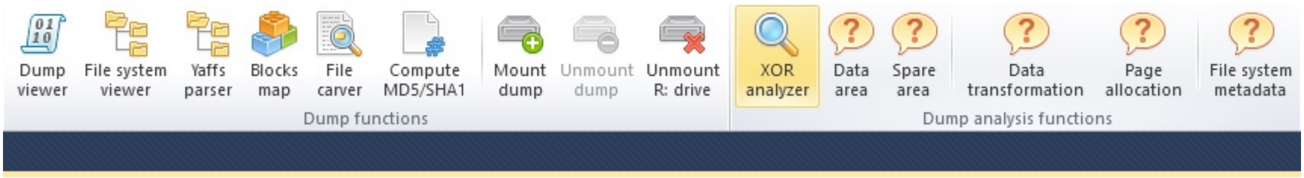
- Data area XOR - The most popular XOR key which is used by Alcor Micro controllers is the one with signature **988EE1** and it commonly appears in cases when the Service area has 2 Bytes.

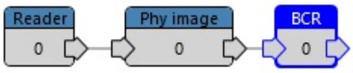


- Service area XOR - Contain adjusted to the page size XOR key, for the first byte of the Service area. In many cases, when SA is XORed, this XOR should be connected in the last place, after XOR for the Data area and after the ECC element.

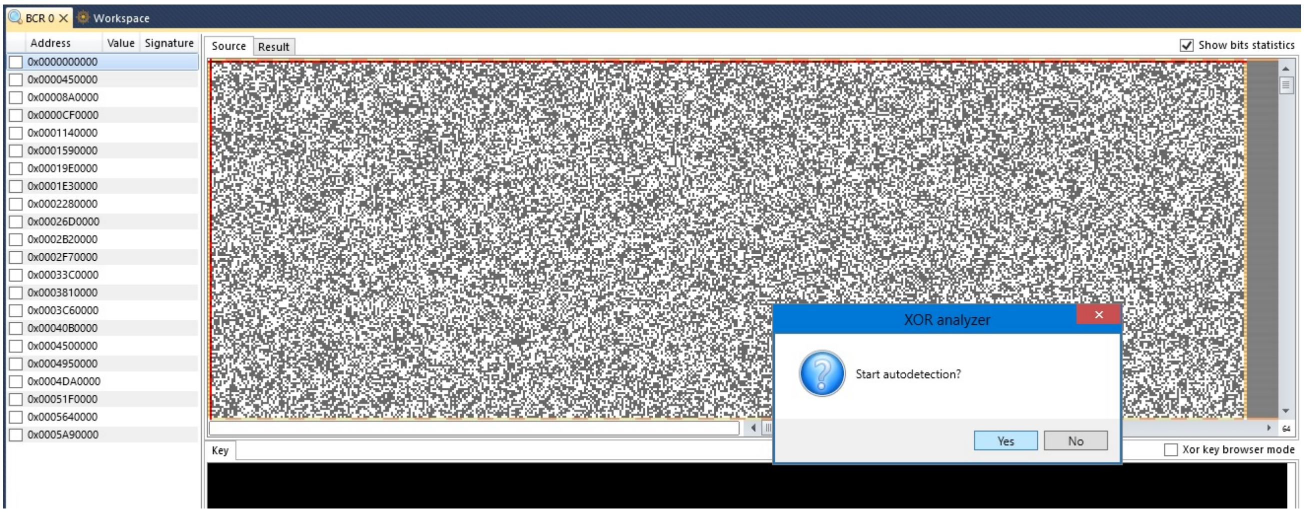
XOR key determination - Data area

To detect the XOR key for Data Area click on the element where you want to apply XOR, it should be Physical Image, ECC, or BCR in case of bad columns. Then run the XOR analyzer.


Dump viewer
File system viewer
Yaffs parser
Blocks map
File carver
Compute MD5/SHA1
Mount dump
Unmount dump
Unmount R: drive
Dump functions
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions


Reader
0
Phy image
0
BCR
0

Start XOR autodetection.

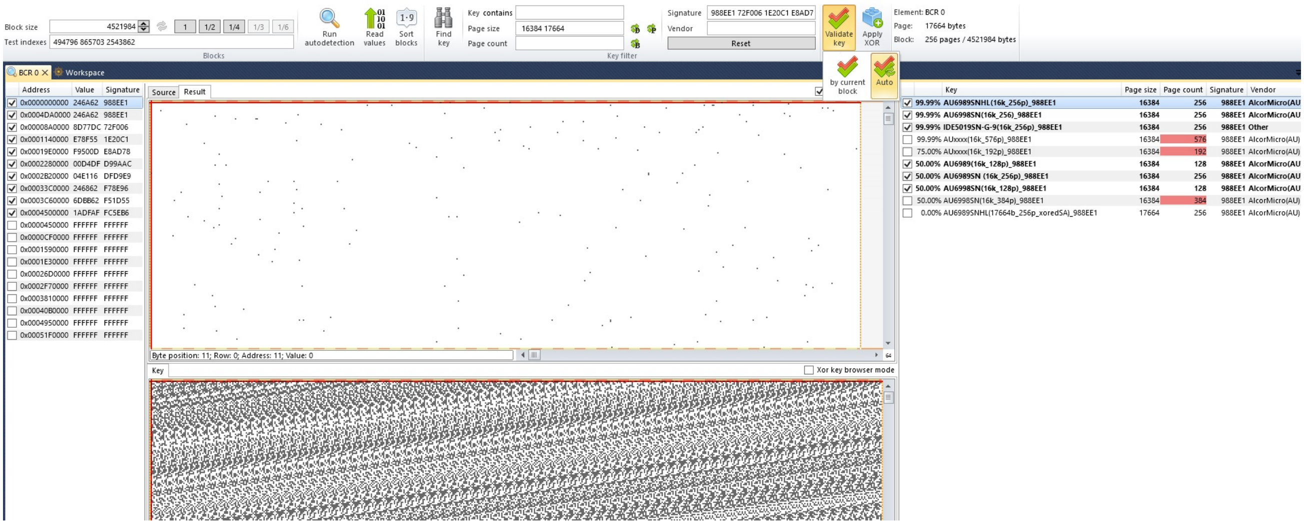

BCR 0
Workspace
Source
Result
Show bits statistics
Key
Xor key browser mode
Yes
No

	Address	Value	Signature
☐	0x0000000000		
☐	0x0000450000		
☐	0x00008A0000		
☐	0x0000CF0000		
☐	0x0001140000		
☐	0x0001590000		
☐	0x00019E0000		
☐	0x0001E30000		
☐	0x0002280000		
☐	0x00026D0000		
☐	0x0002B20000		
☐	0x0002F70000		
☐	0x00033C0000		
☐	0x0003810000		
☐	0x0003C60000		
☐	0x00040B0000		
☐	0x0004500000		
☐	0x0004950000		
☐	0x0004DA0000		
☐	0x00051F0000		
☐	0x0005640000		
☐	0x0005A90000		

XOR analyzer

Start autodetection?

After autodetection, XOR keys purposes will be displayed on the right panel, to check which one is correct, click on the "Validate key" button. In this case, 3 matching keys have been detected and now it's necessary to attach one of them.


1
1/2
1/4
1/3
1/6
Blocks
Run autodetection
Read values
Sort blocks
Find key
Key filter
Reset
Validate key
Apply XOR
by current block
Auto
BCR 0
Workspace
Source
Result
Key
Xor key browser mode

Block size: 4521984

Test indexes: 494796 865703 2543862

Key contains:

Page size: 16384 17664

Page count:

Signature: 988EE1 72F006 1E20C1 E8AD7

Vendor:

Element: BCR 0

Page: 17664 bytes

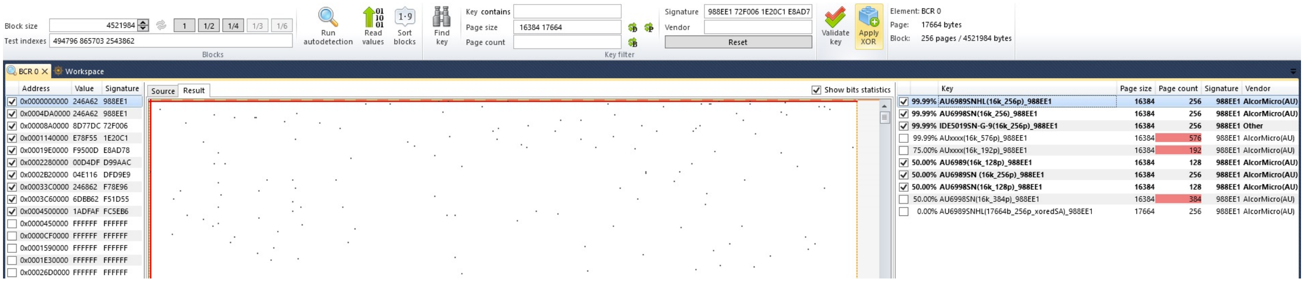
Block: 256 pages / 4521984 bytes

	Address	Value	Signature
☑	0x0000000000	246A62	988EE1
☑	0x0004DA0000	246A62	988EE1
☑	0x00008A0000	8D77DC	72F006
☑	0x0001140000	E78F55	1E20C1
☑	0x00019E0000	F9500D	E8AD78
☑	0x0002280000	00D4DF	D99AAC
☑	0x0002B20000	04E116	DFD9E9
☑	0x00033C0000	246862	F78E96
☑	0x0003C60000	6DBB62	F51D55
☑	0x0004500000	1ADFAF	FC5EB6
☐	0x0000450000	FFFFFF	FFFFFF
☐	0x0000CF0000	FFFFFF	FFFFFF
☐	0x0001590000	FFFFFF	FFFFFF
☐	0x0001E30000	FFFFFF	FFFFFF
☐	0x00026D0000	FFFFFF	FFFFFF
☐	0x0002F70000	FFFFFF	FFFFFF
☐	0x0003810000	FFFFFF	FFFFFF
☐	0x00040B0000	FFFFFF	FFFFFF
☐	0x0004950000	FFFFFF	FFFFFF
☐	0x00051F0000	FFFFFF	FFFFFF

Byte position: 11; Row: 0; Address: 11; Value: 0

		Key	Page size	Page count	Signature	Vendor
☑	**99.99%**	**AU6989SNHL(16k_256p)_988EE1**	**16384**	**256**	**988EE1**	**AlcorMicro(AU**
☑	**99.99%**	**AU6998SN(16k_256)_988EE1**	**16384**	**256**	**988EE1**	**AlcorMicro(AU**
☑	**99.99%**	**IDE5019SN-G-9(16k_256p)_988EE1**	**16384**	**256**	**988EE1**	**Other**
☐	99.99%	AUxxxx(16k_576p)_988EE1	16384	576	988EE1	AlcorMicro(AU)
☐	75.00%	AUxxxx(16k_192p)_988EE1	16384	192	988EE1	AlcorMicro(AU)
☑	**50.00%**	**AU6989(16k_128p)_988EE1**	**16384**	**128**	**988EE1**	**AlcorMicro(AU**
☑	**50.00%**	**AU6989SN (16k_256p)_988EE1**	**16384**	**256**	**988EE1**	**AlcorMicro(AU**
☑	**50.00%**	**AU6998SN(16k_128p)_988EE1**	**16384**	**128**	**988EE1**	**AlcorMicro(AU**
☐	50.00%	AU6998SN(16k_384p)_988EE1	16384	384	988EE1	AlcorMicro(AU)
☐	0.00%	AU6989SNHL(17664b_256p_xoredSA)_988EE1	17664	256	988EE1	AlcorMicro(AU)

To do that, select the XOR key and click "Apply XOR".



Block size: 4521984
Test indexes: 494796 865703 2543862
Key contains:
Page size: 16384 17664
Page count:
Signature: 988EE1 72F006 1E20C1 E8AD7
Vendor:
Element: BCR 0
Page: 17664 bytes
Block: 256 pages / 4521984 bytes

	Address	Value	Signature
☑	0x0000000000	246A62	988EE1
☑	0x0004DA0000	246A62	988EE1
☑	0x00008A0000	8D77DC	72F006
☑	0x0001140000	E78F55	1E20C1
☑	0x00019E0000	F9500D	E8AD78
☑	0x0002280000	00D4DF	D99AAC
☑	0x0002B20000	04E116	DFD9E9
☑	0x00033C0000	246862	F78E96
☑	0x0003C60000	6DBB62	F51D55
☑	0x0004500000	1ADFAF	FC5EB6
☐	0x0000450000	FFFFFF	FFFFFF
☐	0x0000CF0000	FFFFFF	FFFFFF
☐	0x0001590000	FFFFFF	FFFFFF
☐	0x0001E30000	FFFFFF	FFFFFF
☐	0x00026D0000	FFFFFF	FFFFFF

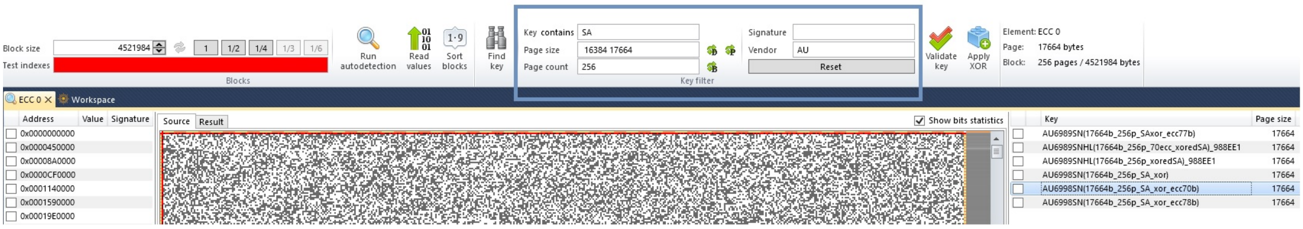
☑ Show bits statistics

		Key	Page size	Page count	Signature	Vendor
☑	**99.99%**	**AU6989SNHL(16k_256p)_988EE1**	**16384**	**256**	**988EE1**	**AlcorMicro(AU)**
☑	**99.99%**	**AU6998SN(16k_256)_988EE1**	**16384**	**256**	**988EE1**	**AlcorMicro(AU)**
☑	**99.99%**	**IDE5019SN-G-9(16k_256p)_988EE1**	**16384**	**256**	**988EE1**	**Other**
☐	99.99%	AUxxxx(16k_576p)_988EE1	16384	576	988EE1	AlcorMicro(AU)
☐	75.00%	AUxxxx(16k_192p)_988EE1	16384	192	988EE1	AlcorMicro(AU)
☑	**50.00%**	**AU6989(16k_128p)_988EE1**	**16384**	**128**	**988EE1**	**AlcorMicro(AU)**
☑	**50.00%**	**AU6989SN (16k_256p)_988EE1**	**16384**	**256**	**988EE1**	**AlcorMicro(AU)**
☑	**50.00%**	**AU6998SN(16k_128p)_988EE1**	**16384**	**128**	**988EE1**	**AlcorMicro(AU)**
☐	50.00%	AU6998SN(16k_384p)_988EE1	16384	384	988EE1	AlcorMicro(AU)
☐	0.00%	AU6989SNHL(17664b_256p_xoredSA)_988EE1	17664	256	988EE1	AlcorMicro(AU)

> ⓘ More details about XOR keys and their detection through XOR analyzer are available in these videos: VNR XOR analyzer , Inversion. Data transformation analysis. XOR. XOR analyzer and in the article XOR.

XOR key determination - Service area

To determine XOR for the Service area open the XOR analyzer, fill out key filters, like on the example below, and then select a proper XOR key according to the size of the ECC area. In our case, the ECC area has 70 Bytes, so XOR for 70 Bytes ECC should be selected.



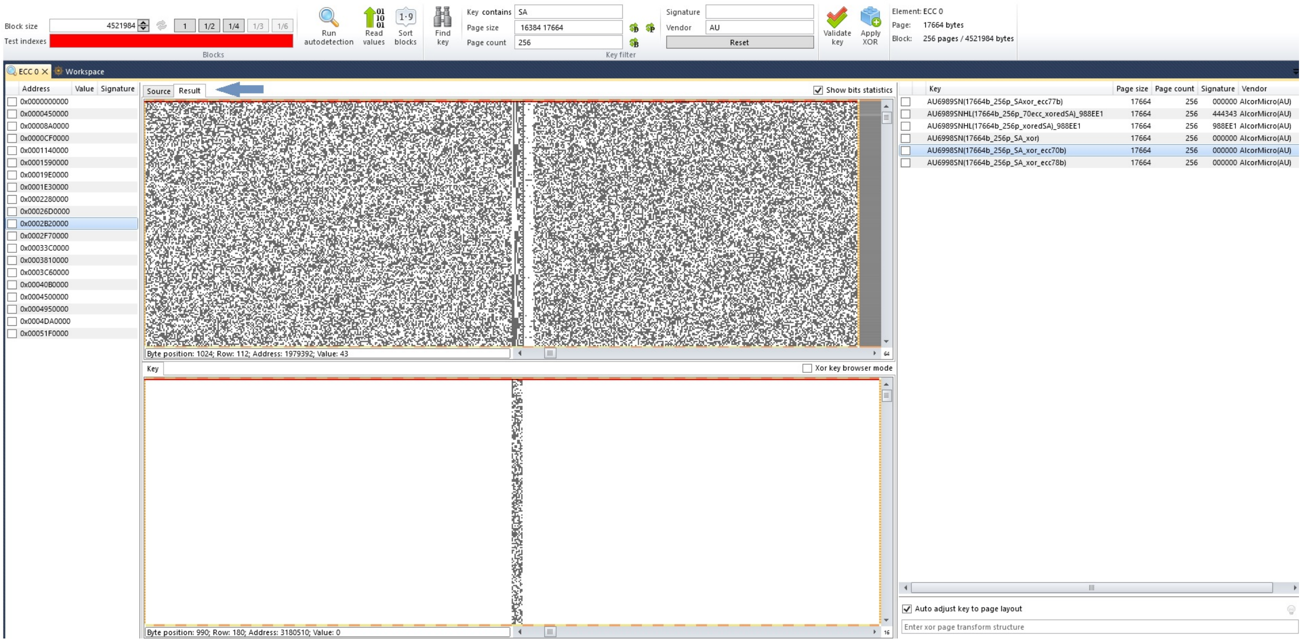
Block size: 4521984
Test indexes:
Key contains: SA
Page size: 16384 17664
Page count: 256
Signature:
Vendor: AU
Element: ECC 0
Page: 17664 bytes
Block: 256 pages / 4521984 bytes

	Address	Value	Signature
☐	0x0000000000		
☐	0x0000450000		
☐	0x00008A0000		
☐	0x0000CF0000		
☐	0x0001140000		
☐	0x0001590000		
☐	0x00019E0000		

☑ Show bits statistics

	Key	Page size
☐	AU6989SN(17664b_256p_SAxor_ecc77b)	17664
☐	AU6989SNHL(17664b_256p_70ecc_xoredSA)_988EE1	17664
☐	AU6989SNHL(17664b_256p_xoredSA)_988EE1	17664
☐	AU6998SN(17664b_256p_SA_xor)	17664
☐	AU6998SN(17664b_256p_SA_xor_ecc70b)	17664
☐	AU6998SN(17664b_256p_SA_xor_ecc78b)	17664

To check if the key is working move to the Service area position and switch tab from "Source" to "Result".



Block size 4521984

Test indexes

Key contains SA

Page size 16384 17664

Page count 256

Signature

Vendor AU

Element: ECC 0

Page: 17664 bytes

Block: 256 pages / 4521984 bytes

	Address	Value	Signature
☐	0x0000000000		
☐	0x0000450000		
☐	0x00008A0000		
☐	0x0000CF0000		
☐	0x0001140000		
☐	0x0001590000		
☐	0x00019E0000		
☐	0x0001E30000		
☐	0x0002280000		
☐	0x00026D0000		
☐	0x0002B20000		
☐	0x0002F70000		
☐	0x00033C0000		
☐	0x0003810000		
☐	0x0003C60000		
☐	0x00040B0000		
☐	0x0004500000		
☐	0x0004950000		
☐	0x0004DA0000		
☐	0x00051F0000		

☑ Show bits statistics

Byte position: 1024; Row: 112; Address: 1979392; Value: 43

☐ Xor key browser mode

Byte position: 990; Row: 180; Address: 3180510; Value: 0

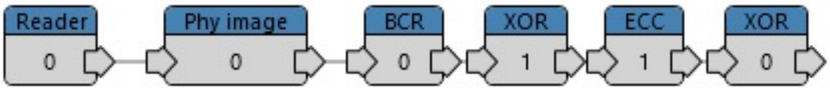
	Key	Page size	Page count	Signature	Vendor
☐	AU6989SN(17664b_256p_SAxor_ecc77b)	17664	256	000000	AlcorMicro(AU)
☐	AU6989SNHL(17664b_256p_70ecc_xoredSA)_988EE1	17664	256	444343	AlcorMicro(AU)
☐	AU6989SNHL(17664b_256p_xoredSA)_988EE1	17664	256	988EE1	AlcorMicro(AU)
☐	AU6998SN(17664b_256p_SA_xor)	17664	256	000000	AlcorMicro(AU)
☐	AU6998SN(17664b_256p_SA_xor_ecc70b)	17664	256	000000	AlcorMicro(AU)
☐	AU6998SN(17664b_256p_SA_xor_ecc78b)	17664	256	000000	AlcorMicro(AU)

☑ Auto adjust key to page layout

Enter xor page transform structure

If we see that the noise pattern has been removed, like on the example above, it means that the key is working and can be attached.

In the majority cases, the XOR key for the Service area will be attached after the XOR key for the Data area and ECC element, like on the example below.

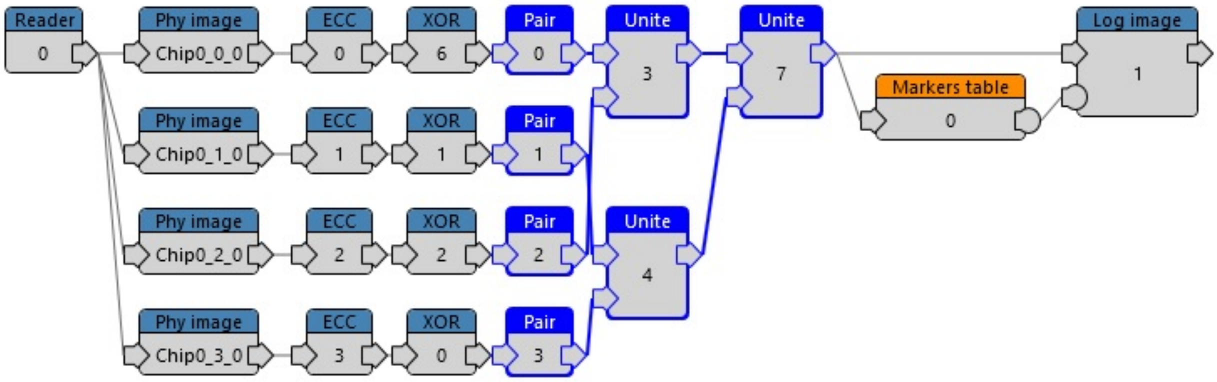


Multi plane page allocation

When a proper ECC and Data transformation has been found and removed in the next step it will be necessary to determine in which mode pages of data were recorded to NAND memory. We can distinguish two types of page allocations:

- Single plane page allocation - sequential page allocation
- Multi plane page allocation - parallel page allocation among physical chips and chip planes of NAND memory.

In case when a device was using Multi plane page allocation is necessary to connect additional **PAIR** element and in the case of multiple chips also the **UNITE** element should be used.

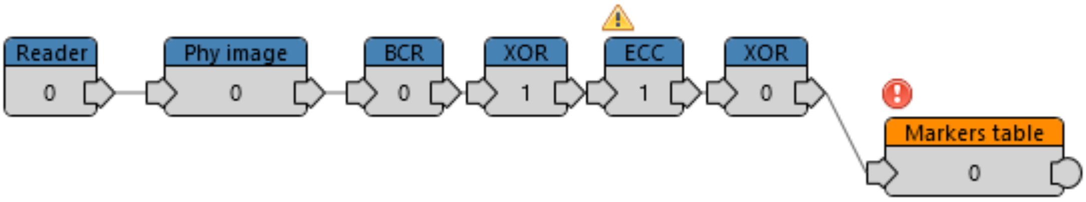

Reader
0
Phy image
Chip0_0_0
ECC
0
XOR
6
Pair
0
Unite
3
Unite
7
Log image
1
Markers table
0
Phy image
Chip0_1_0
ECC
1
XOR
1
Pair
1
Phy image
Chip0_2_0
ECC
2
XOR
2
Pair
2
Unite
4
Phy image
Chip0_3_0
ECC
3
XOR
0
Pair
3

ⓘ Description of those elements, including the procedure of determination controller page allocation mode, is described in the article: Multi Plane Page Allocation and in the video Multi-plane page allocation. PAIR and UNITE usage.

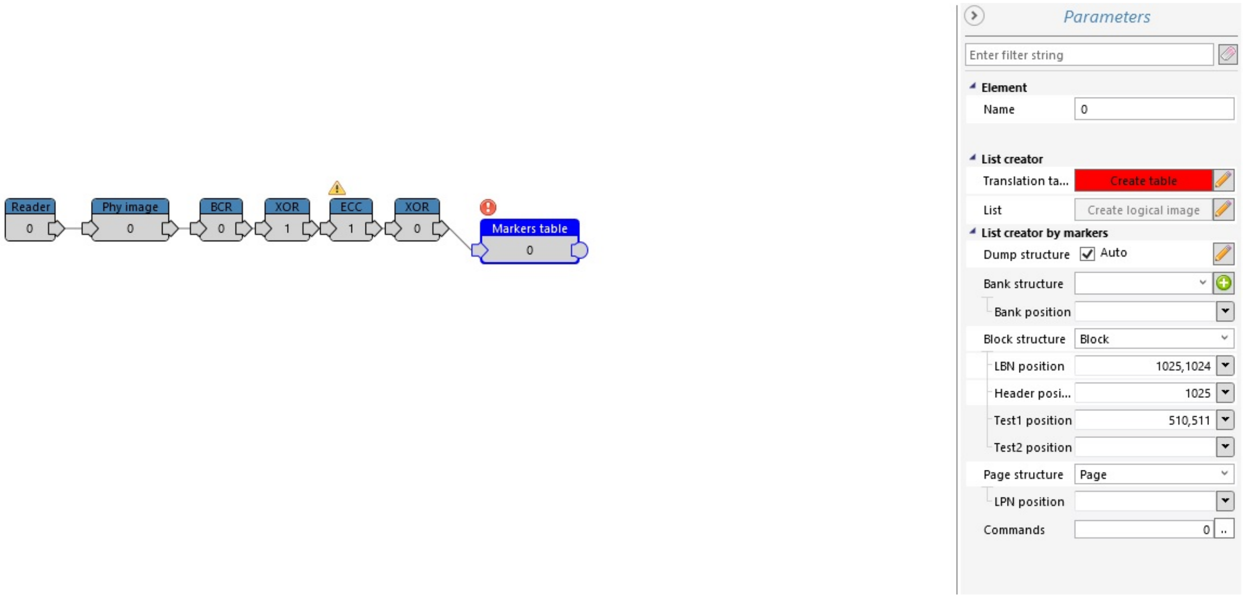
Block management

In the last step of data recovery, it is necessary to assemble blocks of data into a logical array in order to obtain a logical image. It is possible thanks to the Service area where NAND controllers usually write Logical Block Number(LBN) which helps to determine sequence of blocks in the dump.

The Block management step can be processed through the **Markers table**. This element allows to read particular values from each block in the dump, then it is possible to filter and stack these block together according to previously loaded values. To use it, connect the Markers table to the latest element and fill Table indexes with positions of Service area structures (LBN, Header).


Reader
0
Phy image
0
BCR
0
XOR
1
ECC
1
XOR
0
Markers table
0

In Alcor Micro cases, with 2 Bytes Service area, LBN is located on positions [1024, 1025] and it's usually written in the Little-Endian order, therefore in Markers table reversed order of these bytes should be used [1025,1024]. The first Byte of LBN[1025] will be a s a Header and for Test1 we set [510,511], which will help to identify the first block.


Reader
0
Phy image
0
BCR
0
XOR
1
ECC
1
XOR
0
Markers table
0
Parameters
Enter filter string

Element

Name: 0

List creator

Translation ta...: Create table

List: Create logical image

List creator by markers

Dump structure: ☑ Auto

Bank structure:

Bank position:

Block structure: Block

LBN position: 1025,1024

Header posi...: 1025

Test1 position: 510,511

Test2 position:

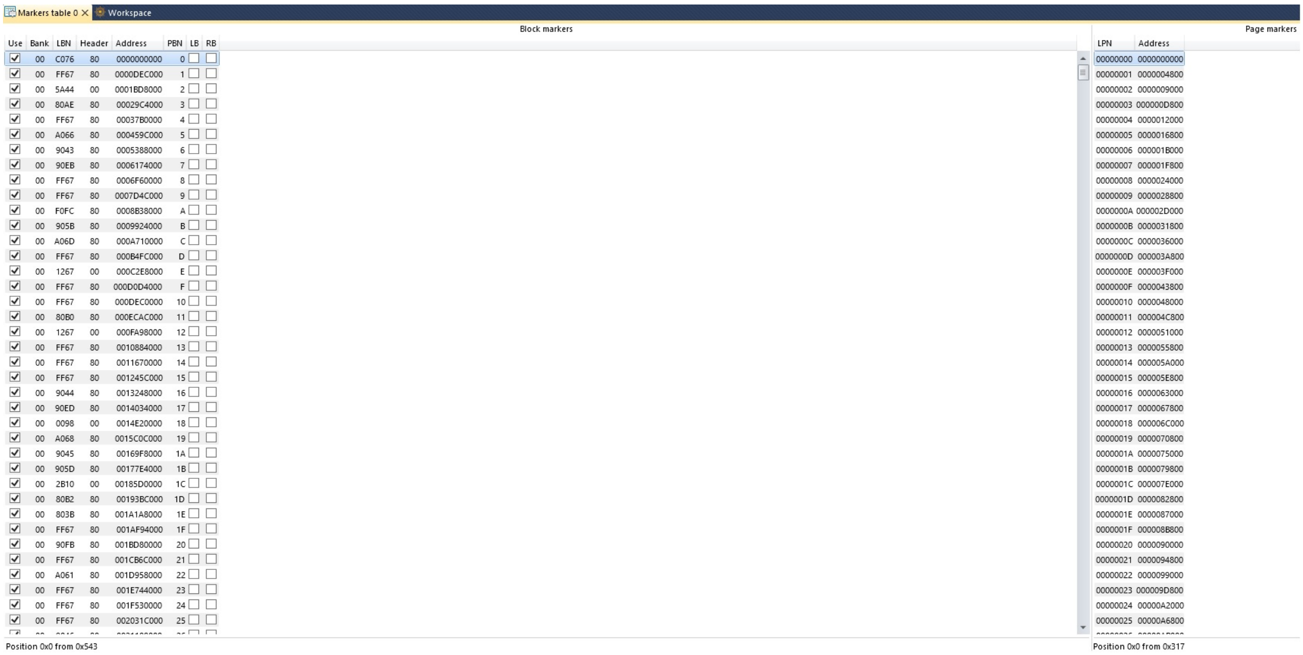
Page structure: Page

LPN position:

Commands: 0

> ⓘ More information about Service area structures, Endianness, and basics of Markers table is presented in the video: Block management. LBN, headers, Marker table.

When a Markers table Indexes have been filled, click on the "Create table" button. Now VNR will load values, from previously selected positions, from each block in the dump. When values have been loaded, the table can be open.



Block markers

Use	Bank	LBN	Header	Address	PBN	LB	RB
☑	00	C076	80	0000000000	0	☐	☐
☑	00	FF67	80	0000DEC000	1	☐	☐
☑	00	5A44	00	0001BD8000	2	☐	☐
☑	00	80AE	80	00029C4000	3	☐	☐
☑	00	FF67	80	00037B0000	4	☐	☐
☑	00	A066	80	000459C000	5	☐	☐
☑	00	9043	80	0005388000	6	☐	☐
☑	00	90EB	80	0006174000	7	☐	☐
☑	00	FF67	80	0006F60000	8	☐	☐
☑	00	FF67	80	0007D4C000	9	☐	☐
☑	00	F0FC	80	0008B38000	A	☐	☐
☑	00	905B	80	0009924000	B	☐	☐
☑	00	A06D	80	000A710000	C	☐	☐
☑	00	FF67	80	000B4FC000	D	☐	☐
☑	00	1267	00	000C2E8000	E	☐	☐
☑	00	FF67	80	000D0D4000	F	☐	☐
☑	00	FF67	80	000DEC0000	10	☐	☐
☑	00	80B0	80	000ECAC000	11	☐	☐
☑	00	1267	00	000FA98000	12	☐	☐
☑	00	FF67	80	0010884000	13	☐	☐
☑	00	FF67	80	0011670000	14	☐	☐
☑	00	FF67	80	001245C000	15	☐	☐
☑	00	9044	80	0013248000	16	☐	☐
☑	00	90ED	80	0014034000	17	☐	☐
☑	00	0098	00	0014E20000	18	☐	☐
☑	00	A068	80	0015C0C000	19	☐	☐
☑	00	9045	80	00169F8000	1A	☐	☐
☑	00	905D	80	00177E4000	1B	☐	☐
☑	00	2B10	00	00185D0000	1C	☐	☐
☑	00	80B2	80	00193BC000	1D	☐	☐
☑	00	803B	80	001A1A8000	1E	☐	☐
☑	00	FF67	80	001AF94000	1F	☐	☐
☑	00	90FB	80	001BD80000	20	☐	☐
☑	00	FF67	80	001CB6C000	21	☐	☐
☑	00	A061	80	001D958000	22	☐	☐
☑	00	FF67	80	001E744000	23	☐	☐
☑	00	FF67	80	001F530000	24	☐	☐
☑	00	FF67	80	002031C000	25	☐	☐

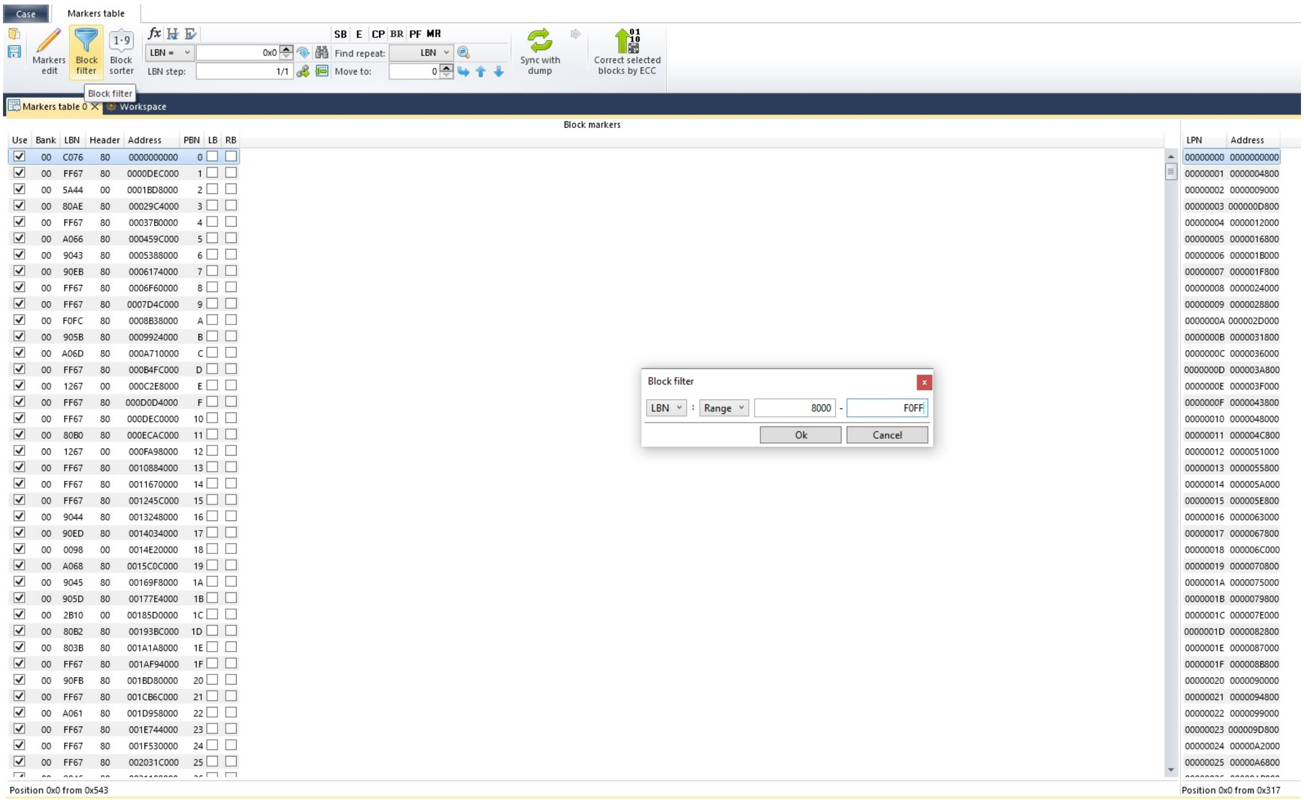
Position 0x0 from 0x543

Page markers

LPN	Address
00000000	0000000000
00000001	0000004800
00000002	0000009000
00000003	000000D800
00000004	0000012000
00000005	0000016800
00000006	000001B000
00000007	000001F800
00000008	0000024000
00000009	0000028800
0000000A	000002D000
0000000B	0000031800
0000000C	0000036000
0000000D	000003A800
0000000E	000003F000
0000000F	0000043800
00000010	0000048000
00000011	000004C800
00000012	0000051000
00000013	0000055800
00000014	000005A000
00000015	000005E800
00000016	0000063000
00000017	0000067800
00000018	000006C000
00000019	0000070800
0000001A	0000075000
0000001B	0000079800
0000001C	000007E000
0000001D	0000082800
0000001E	0000087000
0000001F	000008B800
00000020	0000090000
00000021	0000094800
00000022	0000099000
00000023	000009D800
00000024	00000A2000
00000025	00000A6800

Position 0x0 from 0x317

Now it will be necessary to filter out blocks that are out of the LBN sequence, in Alcor Micro's usually this sequence of blocks starts from 0x8000 and ends at 0xF0FF. To filter LBN by this sequence click on the **"Block filter"** button and filter LBN by the range [8000 - F0FF].



LBN = 0x0

LBN step: 1/1

Find repeat: LBN

Move to: 0

Block filter

LBN : Range 8000 - F0FF

Ok Cancel

Block markers

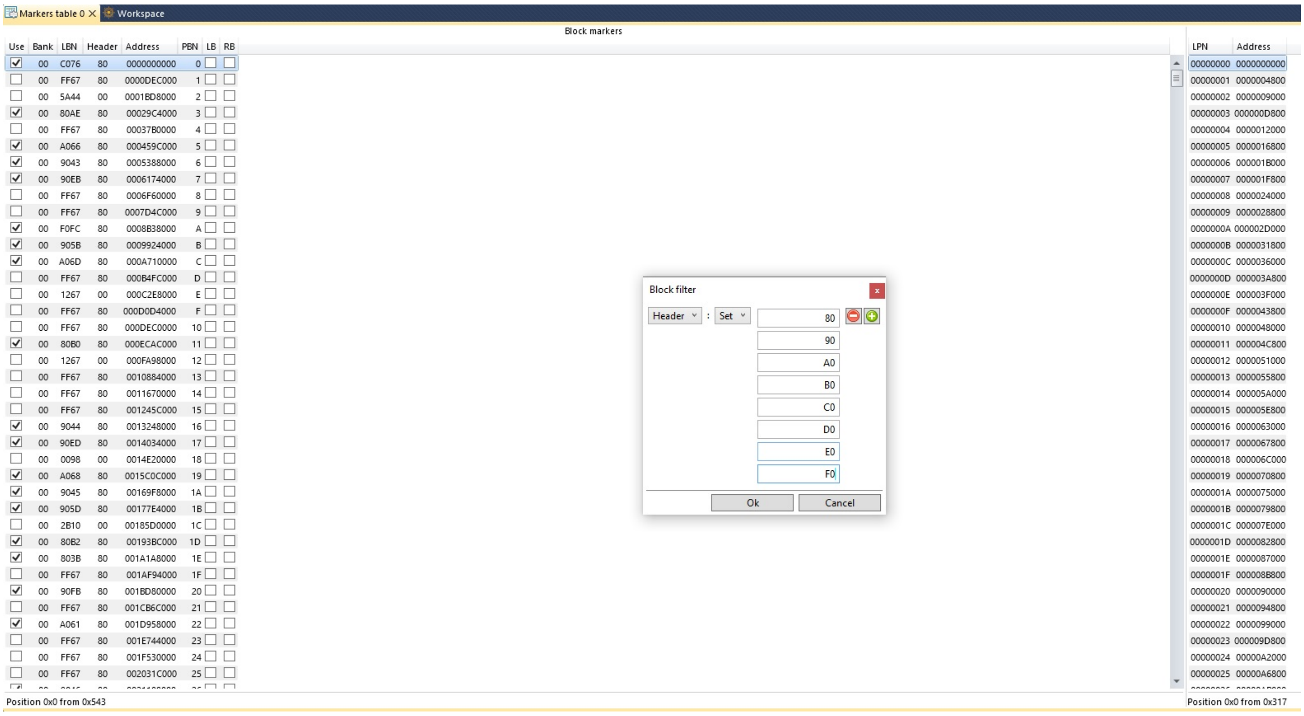
Use	Bank	LBN	Header	Address	PBN	LB	RB
☑	00	C076	80	0000000000	0	☐	☐
☑	00	FF67	80	0000DEC000	1	☐	☐
☑	00	5A44	00	0001BD8000	2	☐	☐
☑	00	80AE	80	00029C4000	3	☐	☐
☑	00	FF67	80	00037B0000	4	☐	☐
☑	00	A066	80	000459C000	5	☐	☐
☑	00	9043	80	0005388000	6	☐	☐
☑	00	90EB	80	0006174000	7	☐	☐
☑	00	FF67	80	0006F60000	8	☐	☐
☑	00	FF67	80	0007D4C000	9	☐	☐
☑	00	F0FC	80	0008B38000	A	☐	☐
☑	00	905B	80	0009924000	B	☐	☐
☑	00	A06D	80	000A710000	C	☐	☐
☑	00	FF67	80	000B4FC000	D	☐	☐
☑	00	1267	00	000C2E8000	E	☐	☐
☑	00	FF67	80	000D0D4000	F	☐	☐
☑	00	FF67	80	000DEC0000	10	☐	☐
☑	00	80B0	80	000ECAC000	11	☐	☐
☑	00	1267	00	000FA98000	12	☐	☐
☑	00	FF67	80	0010884000	13	☐	☐
☑	00	FF67	80	0011670000	14	☐	☐
☑	00	FF67	80	001245C000	15	☐	☐
☑	00	9044	80	0013248000	16	☐	☐
☑	00	90ED	80	0014034000	17	☐	☐
☑	00	0098	00	0014E20000	18	☐	☐
☑	00	A068	80	0015C0C000	19	☐	☐
☑	00	9045	80	00169F8000	1A	☐	☐
☑	00	905D	80	00177E4000	1B	☐	☐
☑	00	2B10	00	00185D0000	1C	☐	☐
☑	00	80B2	80	00193BC000	1D	☐	☐
☑	00	803B	80	001A1A8000	1E	☐	☐
☑	00	FF67	80	001AF94000	1F	☐	☐
☑	00	90FB	80	001BD80000	20	☐	☐
☑	00	FF67	80	001CB6C000	21	☐	☐
☑	00	A061	80	001D958000	22	☐	☐
☑	00	FF67	80	001E744000	23	☐	☐
☑	00	FF67	80	001F530000	24	☐	☐
☑	00	FF67	80	002031C000	25	☐	☐

Position 0x0 from 0x543

LPN	Address
00000000	0000000000
00000001	0000004800
00000002	0000009000
00000003	000000D800
00000004	0000012000
00000005	0000016800
00000006	000001B000
00000007	000001F800
00000008	0000024000
00000009	0000028800
0000000A	000002D000
0000000B	0000031800
0000000C	0000036000
0000000D	000003A800
0000000E	000003F000
0000000F	0000043800
00000010	0000048000
00000011	000004C800
00000012	0000051000
00000013	0000055800
00000014	000005A000
00000015	000005E800
00000016	0000063000
00000017	0000067800
00000018	000006C000
00000019	0000070800
0000001A	0000075000
0000001B	0000079800
0000001C	000007E000
0000001D	0000082800
0000001E	0000087000
0000001F	000008B800
00000020	0000090000
00000021	0000094800
00000022	0000099000
00000023	000009D800
00000024	00000A2000
00000025	00000A6800

Position 0x0 from 0x317

Then the Header structure should be also filtered, with the "Set" rule by: [80,90,A0,B0,C0,D0,E0,F0], like on the example below.



Use	Bank	LBN	Header	Address	PBN	LB	RB
☑	00	C076	80	0000000000	0	☐	☐
☐	00	FF67	80	0000DEC000	1	☐	☐
☐	00	5A44	00	0001BD8000	2	☐	☐
☑	00	80AE	80	00029C4000	3	☐	☐
☐	00	FF67	80	00037B0000	4	☐	☐
☑	00	A066	80	000459C000	5	☐	☐
☑	00	9043	80	0005388000	6	☐	☐
☑	00	90EB	80	0006174000	7	☐	☐
☐	00	FF67	80	0006F60000	8	☐	☐
☐	00	FF67	80	0007D4C000	9	☐	☐
☑	00	F0FC	80	0008B38000	A	☐	☐
☑	00	905B	80	0009924000	B	☐	☐
☑	00	A06D	80	000A710000	C	☐	☐
☐	00	FF67	80	000B4FC000	D	☐	☐
☐	00	1267	00	000C2E8000	E	☐	☐
☐	00	FF67	80	000D0D4000	F	☐	☐
☐	00	FF67	80	000DEC0000	10	☐	☐
☑	00	80B0	80	000ECAC000	11	☐	☐
☐	00	1267	00	000FA98000	12	☐	☐
☐	00	FF67	80	0010884000	13	☐	☐
☐	00	FF67	80	0011670000	14	☐	☐
☐	00	FF67	80	001245C000	15	☐	☐
☑	00	9044	80	0013248000	16	☐	☐
☑	00	90ED	80	0014034000	17	☐	☐
☐	00	0098	00	0014E20000	18	☐	☐
☑	00	A068	80	0015C0C000	19	☐	☐
☑	00	9045	80	00169F8000	1A	☐	☐
☑	00	905D	80	00177E4000	1B	☐	☐
☐	00	2B10	00	00185D0000	1C	☐	☐
☑	00	80B2	80	00193BC000	1D	☐	☐
☑	00	803B	80	001A1A8000	1E	☐	☐
☐	00	FF67	80	001AF94000	1F	☐	☐
☑	00	90FB	80	001BD80000	20	☐	☐
☐	00	FF67	80	001CB6C000	21	☐	☐
☑	00	A061	80	001D958000	22	☐	☐
☐	00	FF67	80	001E744000	23	☐	☐
☐	00	FF67	80	001F530000	24	☐	☐
☐	00	FF67	80	002031C000	25	☐	☐

Position 0x0 from 0x543

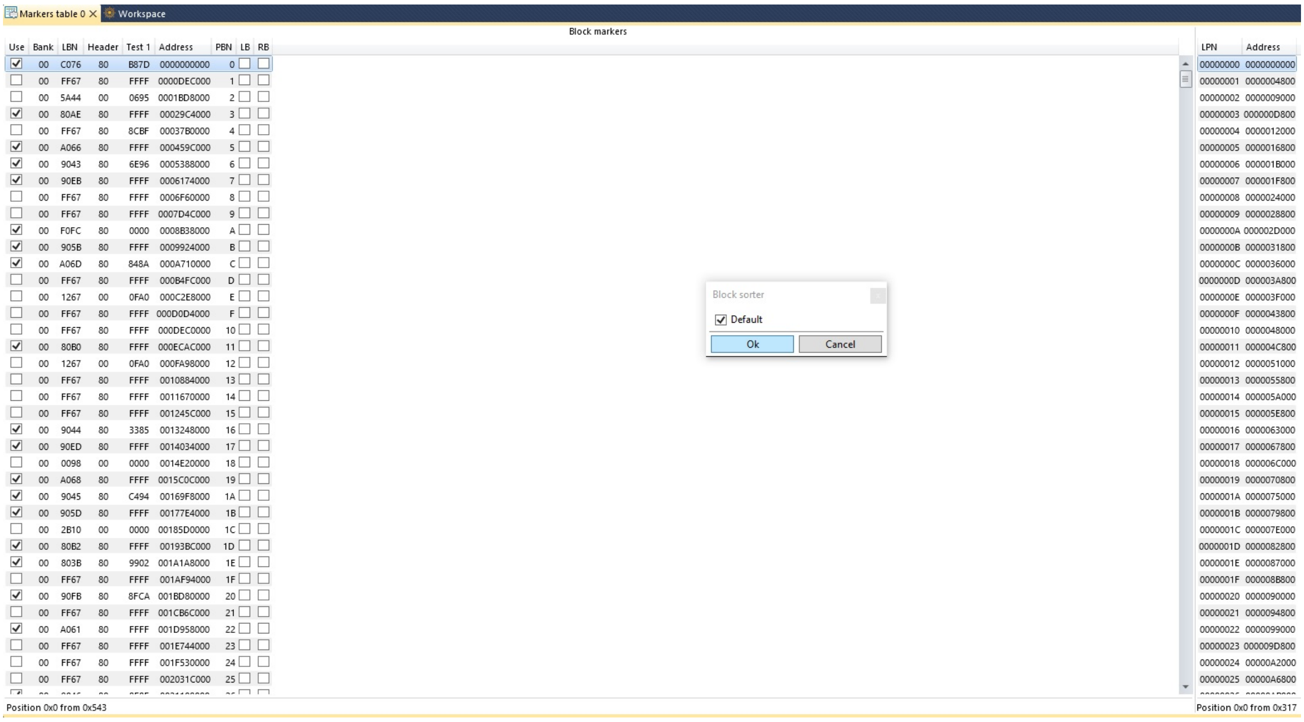
LPN	Address
00000000	0000000000
00000001	0000004800
00000002	0000009000
00000003	000000D800
00000004	0000012000
00000005	0000016800
00000006	000001B000
00000007	000001F800
00000008	0000024000
00000009	0000028800
0000000A	000002D000
0000000B	0000031800
0000000C	0000036000
0000000D	000003A800
0000000E	000003F000
0000000F	0000043800
00000010	0000048000
00000011	000004C800
00000012	0000051000
00000013	0000055800
00000014	000005A000
00000015	000005E800
00000016	0000063000
00000017	0000067800
00000018	000006C000
00000019	0000070800
0000001A	0000075000
0000001B	0000079800
0000001C	000007E000
0000001D	0000082800
0000001E	0000087000
0000001F	000008B800
00000020	0000090000
00000021	0000094800
00000022	0000099000
00000023	000009D800
00000024	00000A2000
00000025	00000A6800

Position 0x0 from 0x317

Block filter

Header : Set 80 90 A0 B0 C0 D0 E0 F0

When blocks have been filtered, in the next step sequence of blocks should be examined. At first is necessary to sort blocks.



Use	Bank	LBN	Header	Test 1	Address	PBN	LB	RB
☑	00	C076	80	B87D	0000000000	0	☐	☐
☐	00	FF67	80	FFFF	0000DEC000	1	☐	☐
☐	00	5A44	00	0695	0001BD8000	2	☐	☐
☑	00	80AE	80	FFFF	00029C4000	3	☐	☐
☐	00	FF67	80	8CBF	00037B0000	4	☐	☐
☑	00	A066	80	FFFF	000459C000	5	☐	☐
☑	00	9043	80	6E96	0005388000	6	☐	☐
☑	00	90EB	80	FFFF	0006174000	7	☐	☐
☐	00	FF67	80	FFFF	0006F60000	8	☐	☐
☐	00	FF67	80	FFFF	0007D4C000	9	☐	☐
☑	00	F0FC	80	0000	0008B38000	A	☐	☐
☑	00	905B	80	FFFF	0009924000	B	☐	☐
☑	00	A06D	80	848A	000A710000	C	☐	☐
☐	00	FF67	80	FFFF	000B4FC000	D	☐	☐
☐	00	1267	00	0FA0	000C2E8000	E	☐	☐
☐	00	FF67	80	FFFF	000D0D4000	F	☐	☐
☐	00	FF67	80	FFFF	000DEC0000	10	☐	☐
☑	00	80B0	80	FFFF	000ECAC000	11	☐	☐
☐	00	1267	00	0FA0	000FA98000	12	☐	☐
☐	00	FF67	80	FFFF	0010884000	13	☐	☐
☐	00	FF67	80	FFFF	0011670000	14	☐	☐
☐	00	FF67	80	FFFF	001245C000	15	☐	☐
☑	00	9044	80	3385	0013248000	16	☐	☐
☑	00	90ED	80	FFFF	0014034000	17	☐	☐
☐	00	0098	00	0000	0014E20000	18	☐	☐
☑	00	A068	80	FFFF	0015C0C000	19	☐	☐
☑	00	9045	80	C494	00169F8000	1A	☐	☐
☑	00	905D	80	FFFF	00177E4000	1B	☐	☐
☐	00	2B10	00	0000	00185D0000	1C	☐	☐
☑	00	80B2	80	FFFF	00193BC000	1D	☐	☐
☑	00	803B	80	9902	001A1A8000	1E	☐	☐
☐	00	FF67	80	FFFF	001AF94000	1F	☐	☐
☑	00	90FB	80	8FCA	001BD80000	20	☐	☐
☐	00	FF67	80	FFFF	001CB6C000	21	☐	☐
☑	00	A061	80	FFFF	001D958000	22	☐	☐
☐	00	FF67	80	FFFF	001E744000	23	☐	☐
☐	00	FF67	80	FFFF	001F530000	24	☐	☐
☐	00	FF67	80	FFFF	002031C000	25	☐	☐

Position 0x0 from 0x543

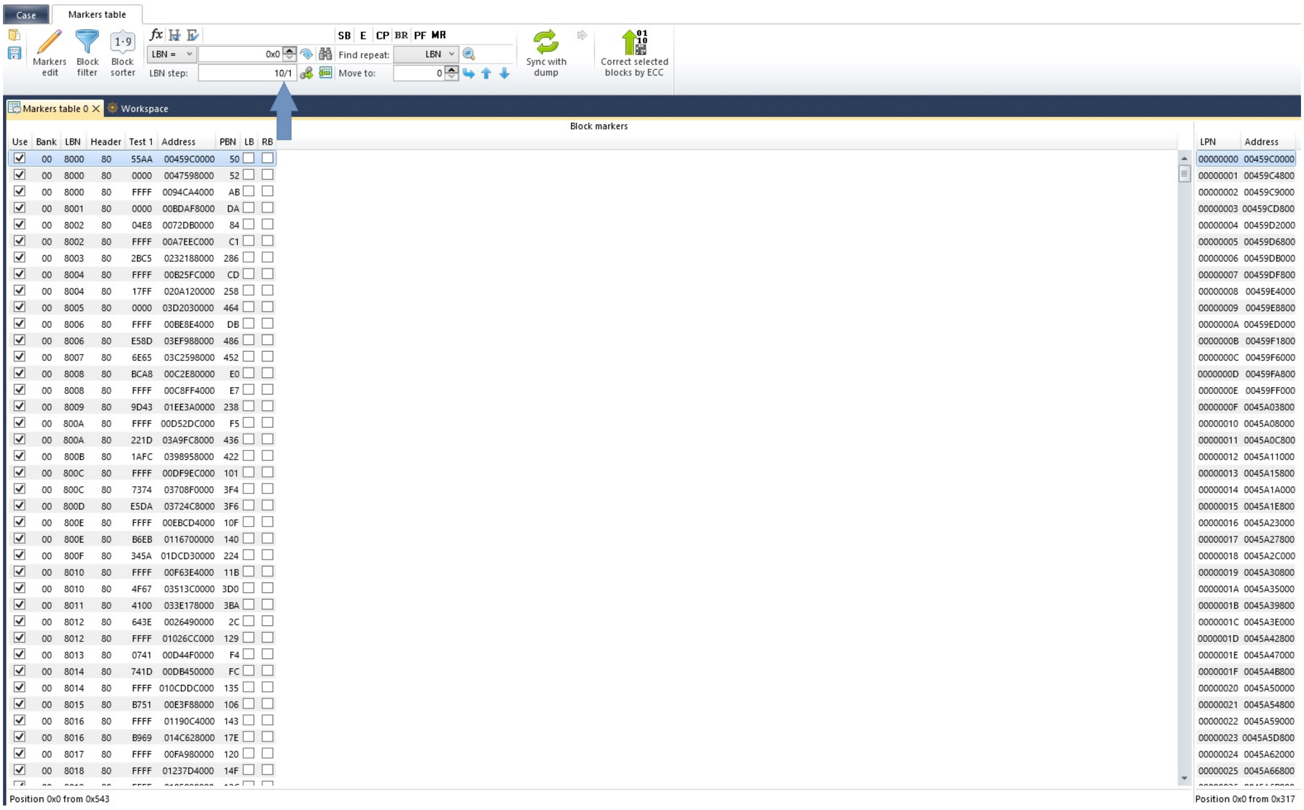
LPN	Address
00000000	0000000000
00000001	0000004800
00000002	0000009000
00000003	000000D800
00000004	0000012000
00000005	0000016800
00000006	000001B000
00000007	000001F800
00000008	0000024000
00000009	0000028800
0000000A	000002D000
0000000B	0000031800
0000000C	0000036000
0000000D	000003A800
0000000E	000003F000
0000000F	0000043800
00000010	0000048000
00000011	000004C800
00000012	0000051000
00000013	0000055800
00000014	000005A000
00000015	000005E800
00000016	0000063000
00000017	0000067800
00000018	000006C000
00000019	0000070800
0000001A	0000075000
0000001B	0000079800
0000001C	000007E000
0000001D	0000082800
0000001E	0000087000
0000001F	000008B800
00000020	0000090000
00000021	0000094800
00000022	0000099000
00000023	000009D800
00000024	00000A2000
00000025	00000A6800

Position 0x0 from 0x317

Block sorter

☑ Default

After that LBN step should be changed from 1/1 to 10/1, duplicated blocks should be removed and missing ones should be added. In order to locate a proper first block with the Master Boot Record is necessary to follow values from Test1 and find the MBR signature. For the file systems from the FAT family and NTFS, the signature will be always equal to 0x55AA.


Case
Markers table
Markers edit
Block filter
Block sorter
LBN =
0x0
LBN step:
10/1
SB E CP BR PF MR
Find repeat:
LBN
Move to:
0
Sync with dump
Correct selected blocks by ECC
Markers table 0
Workspace
Block markers

Use	Bank	LBN	Header	Test 1	Address	PBN	LB	RB
☑	00	8000	80	55AA	00459C0000	50	☐	☐
☑	00	8000	80	0000	0047598000	52	☐	☐
☑	00	8000	80	FFFF	0094CA4000	AB	☐	☐
☑	00	8001	80	0000	00BDAF8000	DA	☐	☐
☑	00	8002	80	04E8	0072DB0000	84	☐	☐
☑	00	8002	80	FFFF	00A7EEC000	C1	☐	☐
☑	00	8003	80	2BC5	0232188000	286	☐	☐
☑	00	8004	80	FFFF	00B25FC000	CD	☐	☐
☑	00	8004	80	17FF	020A120000	258	☐	☐
☑	00	8005	80	0000	03D2030000	464	☐	☐
☑	00	8006	80	FFFF	00BE8E4000	DB	☐	☐
☑	00	8006	80	E58D	03EF988000	486	☐	☐
☑	00	8007	80	6E65	03C2598000	452	☐	☐
☑	00	8008	80	BCA8	00C2E80000	E0	☐	☐
☑	00	8008	80	FFFF	00C8FF4000	E7	☐	☐
☑	00	8009	80	9D43	01EE3A0000	238	☐	☐
☑	00	800A	80	FFFF	00D52DC000	F5	☐	☐
☑	00	800A	80	221D	03A9FC8000	436	☐	☐
☑	00	800B	80	1AFC	0398958000	422	☐	☐
☑	00	800C	80	FFFF	00DF9EC000	101	☐	☐
☑	00	800C	80	7374	03708F0000	3F4	☐	☐
☑	00	800D	80	E5DA	03724C8000	3F6	☐	☐
☑	00	800E	80	FFFF	00EBCD4000	10F	☐	☐
☑	00	800E	80	B6EB	0116700000	140	☐	☐
☑	00	800F	80	345A	01DCD30000	224	☐	☐
☑	00	8010	80	FFFF	00F63E4000	11B	☐	☐
☑	00	8010	80	4F67	03513C0000	3D0	☐	☐
☑	00	8011	80	4100	033E178000	3BA	☐	☐
☑	00	8012	80	643E	0026490000	2C	☐	☐
☑	00	8012	80	FFFF	01026CC000	129	☐	☐
☑	00	8013	80	0741	00D44F0000	F4	☐	☐
☑	00	8014	80	741D	00DB450000	FC	☐	☐
☑	00	8014	80	FFFF	010CDDC000	135	☐	☐
☑	00	8015	80	B751	00E3F88000	106	☐	☐
☑	00	8016	80	FFFF	01190C4000	143	☐	☐
☑	00	8016	80	B969	014C628000	17E	☐	☐
☑	00	8017	80	FFFF	00FA980000	120	☐	☐
☑	00	8018	80	FFFF	01237D4000	14F	☐	☐

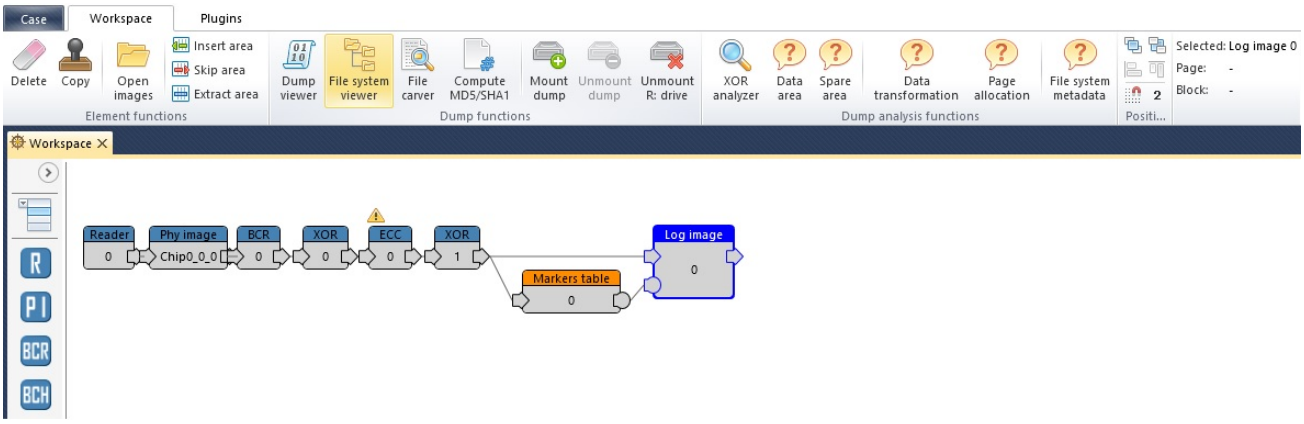
Position 0x0 from 0x543

LPN	Address
00000000	00459C0000
00000001	00459C4800
00000002	00459C9000
00000003	00459CD800
00000004	00459D2000
00000005	00459D6800
00000006	00459DB000
00000007	00459DF800
00000008	00459E4000
00000009	00459E8800
0000000A	00459ED000
0000000B	00459F1800
0000000C	00459F6000
0000000D	00459FA800
0000000E	00459FF000
0000000F	0045A03800
00000010	0045A08000
00000011	0045A0C800
00000012	0045A11000
00000013	0045A15800
00000014	0045A1A000
00000015	0045A1E800
00000016	0045A23000
00000017	0045A27800
00000018	0045A2C000
00000019	0045A30800
0000001A	0045A35000
0000001B	0045A39800
0000001C	0045A3E000
0000001D	0045A42800
0000001E	0045A47000
0000001F	0045A4B800
00000020	0045A50000
00000021	0045A54800
00000022	0045A59000
00000023	0045A5D800
00000024	0045A62000
00000025	0045A66800

Position 0x0 from 0x317

> ⓘ Explanation of how to remove properly duplicated blocks and why is necessary to insert missing ones in Markers table is presented in the video **"Block management. LBN, headers, Marker table" - 1:13:21**

When duplicated blocks have been removed and the gaps have been filled it is possible to build a Logical image. Click **"Create logical image"** in the Markers table parameters tab and connect the **Logical image** element, after that run **File System viewer** from the Workspace panel.


Case
Workspace
Plugins
Delete
Copy
Open images
Insert area
Skip area
Extract area
Element functions
Dump viewer
File system viewer
File carver
Compute MD5/SHA1
Mount dump
Unmount dump
Unmount R: drive
Dump functions
XOR analyzer
Data area
Spare area
Data transformation
Page allocation
File system metadata
Dump analysis functions
2
Positi...
Selected: Log image 0
Page: -
Block: -
Workspace
Reader
0
Phy image
Chip0_0_0
BCR
0
XOR
0
ECC
0
XOR
1
Markers table
0
Log image
0
R
PI
BCR
BCH

https://support.rusolut.com/portal/en/kb/articles/data-recovery-peculiarities-alcor-micro-au-controllers